DRAGON USER

International edition

The independent Dragon magazine

75p US$3.25 February 1984



Graphics animation

Getting down to business

Play Tracker, Drag Racing and Chess

Educational software under review

DOUBLE CHANCE DOUBLE PRIZE

Two sets of extension modules from JCB Microsystems

DRAGON USER

Telephone number
(All departments)
01-437 4343
Editor
GRAHAM CUNNINGHAM
Assistant Editor
ANDREW BRIGGS
Software Editor
GRAHAM TAYLOR
Editorial Secretary
CLEO CHERRY
Advertisement Manager
DAVID LAKE
Advertisement Executive
SIMON LANGSTON
Administration
THERESA LACY
Managing Editor
DUNCAN SCOT
Publishing Director
JENNY IRELAND

Subscriptions
UK £8 for 12 issues
Overseas £14 for 12 issues
ISSN 0265-0177
Dragon User, 12/13 Little Newport Street, London WC2R 3LD.
US address: c/o Business Press International, 205 East 42nd St, New York, NY 10017.
Published by Sunshine Books, Scot Press Ltd. ©Sunshine Books 1984.
Typesetting by Chesham Press, Chesham, Bucks. Printed by Eden Fisher (Southend) Ltd, Southend-on-Sea, Essex.
Distributed by S.M. Distribution, London SW9. 01-274 8611. Telex: 261643
Registered at the Post Office as a newspaper.

How to submit articles

The quality of the material we can publish in *Dragon User* each month will, to a very great extend, depend on the quality of the discoveries that you can make with your Dragon. The Dragon 32 computer was launched on to the market with a powerful version of Basic, but with very poor documentation.

Every one of us who uses a Dragon will be able to discover new tricks and quirks almost every day. To help other Dragon users keep up with the speed of the development each of us must assume that we made the discovery first — that means writing it down and passing it on to others.

Articles which are submitted to *Dragon User* for publication should not be more than 3000 words long. All submissions should be typed. Please leave wide margins and a double space between each line. Programs should, whenever possible, be computer printed on plain white paper and be accompanied by a tape of the program.

We cannot guarantee to return every submitted article or program, so please keep a copy. If you want to have your program returned you must include a stamped, addressed envelope.

Contents

Editorial

FIRST THE BAD news: we've had to raise the cover price of *Dragon User* from 60p to 75p. The good news is that the subscription rate (see the card bound into the back of the magazine) is still the same — £8.00 for 12 issues mailed direct to you. But from the next issue onwards the subscription rate will also have to go up — to £10.00 for 12 issues. So the message is subscribe now if you want to save £2.00.

But the savings don't end there, as a look at the contents of this issue will show you. We've had a lot of correspondence on educational and business software, so this month we've tackled both these subjects. Mike Harrison, a teacher himself, selects the best educational packages while we offer Ged Mead's Dragonsnap, a simple maths game in which two children compete to be the first to spot the answer to sums displayed on the screen. To keep things lively there's also a reward — Ged shows you how to incorporate an arcade-style game into Dragonsnap so that the winner gets a few minutes of fun. On the business side Margaret Norman's Addfile shows you how to write a program which any small business will find useful. Understanding how Addfile works will also enable you to reach a better decision if you're considering buying a bigger business package to use in the office.

And for games players, there's Tracker, where you see if you can do British Rail's job any better — it's ideal for incorporating into Ged's Dragonsnap. And if you want to improve your games programming, take a look at Dave Windle's introduction to the basics of animation. More advanced users have Pam D'Arcy's Tapescan to explore — this machine code program allows recovery from input/output errors giving faster tape positioning. And if it's hardware projects you're after, we show you how to build your own analogue to digital interface — complete with circuit diagrams and all the illustrations you need for the connections.

This is the variety we aim to offer in every issue — although our usual software reviewer will be back next month looking at the latest games for the Dragon (and preparing for a utilities special soon). We think *Dragon User* is a good buy even at 75p — but if you disagree, write and let us know what you think we should be doing.

A rhyme in time

HERE IS a poetical summary of *Dragon User*:

The name is *Dragon User*
For the computer that breathes fire,
Filled with all the articles and software
That an owner could desire.

It even has a problems page
For newcomers to the sport,
Interest for every age
And reviews of games where aliens are fought.

As for external contributions,
Only the best will the Editor use.
Try your hand! I expect he would like some.
I did! and I know he doesn't always refuse.

Now for the readers' programs there's Open File;
To see your work in print must bring a smile.
And for the letters page I say give a cheer
For without it this poet's work wouldn't be here.

Mike Roe,
Essex.

Golf loses handicap

MAY I, through your magazine, contact the many people who have written to me concerning my Golf program published in the July issue of *Dragon User*. I'm afraid there were one or two bugs in my original listing, for which I can only apologise and try to make amends by detailing the cures.

1. Line 9045 had got itself tagged on to the end of line 9040. It should of course be entered as a separate line.
2. Line 5210 should read . . . OR PPOINT (BX, BY+1) = 3 THEN 5190
3. Line 7590 should read . . . : BG = PPOINT (BX, BY+1)
4. If the ball stops on the edge of the hole, it treats the hole as a water hazard. This is cured by adding line 1065 IF GF = 1 THEN 1080
5. The ball may occasionally disappear when it is in a bunker. The cure is to amend line 2130 by inserting PSET (BX, BY, UH) : PSET(BX, BY+1, LH): between THEN and GOTO
6. When the ball is on the green there are times when it and the figure are re-positioned, apparently at random. Amend line 1090 by inserting GF = 0 AND between IF and SQR.

Finally, users without joysticks may like to try amending this listing. Delete line 1120 to 1150 inclusive and substitute:

```
1120 XI = (PEEK(343) = 223) — (PEEK (344) = 223)
1130 YJ = (PEEK(341) = 223) — (PEEK(342) = 223)
1140 IF PEEK (337) = 159 THEN XI = 5*XI:YI = 5*YI
1260 IF PEEK (345) = 223 THEN 1600
1810 IF PEEK (345) = 223 THEN 1730
```

The figure will now move in response to the cursor control keys. Pressing the shift key will increase the distance moved at each step. The backswing is begun by pressing and holding the space bar, and ended by releasing it.

Phil Brookes,
Leominster,
Herefordshire.

Write on Microdeal

HAVING PURCHASED a Dragon primarily to use as a word processor after seeing an ad for Telewriter, I was most interested to read John Scriven's article "A look at the serious side of the Dragon".

The feature of missing out odd letters, caused apparently in all word processing packages by the Dragon's way of scanning its keyboard, did originally slow down typing considerably, but in fairness to Microdeal their latest modification to the Telewriter program, only just issued, has improved things spectacularly and it is now possible to type quite fast.

One of many excellent features of Telewriter is the ease with which it enables one to access characters available on one's printer but not on the Dragon's keyboard — particularly useful if, like me, you have to type a lot of stuff in foreign languages with accents and cedillas.

R Hadekel,
London SW6.

Hi-res input

IF YOU want to input information while using the high resolution screens, then try the following:

```
100 A$ = ""
110 POKE 135,0
120 A = PEEK (135):IF A = 0
    THEN 120
130 IF A = 13 THEN 150
140 A$ = A$ + RIGHT$
    ((CHR$(A)),1): GOTO 100
150 Now proceed with the
    program using the input
    information held in A$
```

If a variable is required then make 150 A = VAL(A$).

M Fecher,
Maidenhead.

Adding to Connect 4

ONE OF the best games you have published in your magazine was Connect 4 in the October issue.

I play it all the time, but found that the coding to ensure that there are no obvious moves for the computer to cover in a vertical direction were missing.

I remedied this by adding the following lines:

```
1141 A=F8
1142 A=A+1:IF A > 6 THEN
     1144
1143 IF GO(I,A)=1 THEN
     SV=SV+1: GOTO 1142
1144 A=F8
1145 A=A-1 : IF A < 1 THEN
     1147
1146 IF GO(I,A)=1 THEN
     SV=SV+1 : GOTO 1145
1147 IF SV > 2 THEN
     T(I)=T(I)+W5
1148 IF SV > 3 THEN
     T(I)=T(I)+W6
```

This has greatly improved the program and makes the computer much harder to beat.

R Davenport,
Cardiff.

Just not cricket

I HAVE just purchased a Dragon 32, and with it Dragon Chess from Oasis Software — and I might add I am very pleased with both. But (and there is always a but isn't there?) A cannot beat the computer.

Although at the moment I do not know much about programs I do know how to play chess. But when I get the computer on the run it calls a draw and stops the game — which just isn't cricket (I mean chess) is it? I like to finish my game even if losing.

The reason I have written to you is to ask if you or any readers can come up with something to override this decision.

Maurice Brown,
Prescot,
Merseyside.

SOUNDS like a professional foul to us. Either your Dragon is the first with artificial intelligence — or there's a fault in the software. Try writing to Oasis Software, Lower North St, Cheddar, Somerset, they should be able to help.

Atari interface

IN RESPONSE to your answer to Stephen Wood in your December issue, there's no need to send off to the us for an interface connecting two Atari-type joysticks to the Dragon.

Cotswold Computers can supply such an interface at £14.95. We also supply Wico's Famous Red Ball joystick at £24.95 and Trackball at £35.95 (Trackball does not need an adaptor).

David Thomlinson,
Cotswold Computers,
6 Middle Row,
Chipping Norton,
Oxfordshire.

Software Top 10

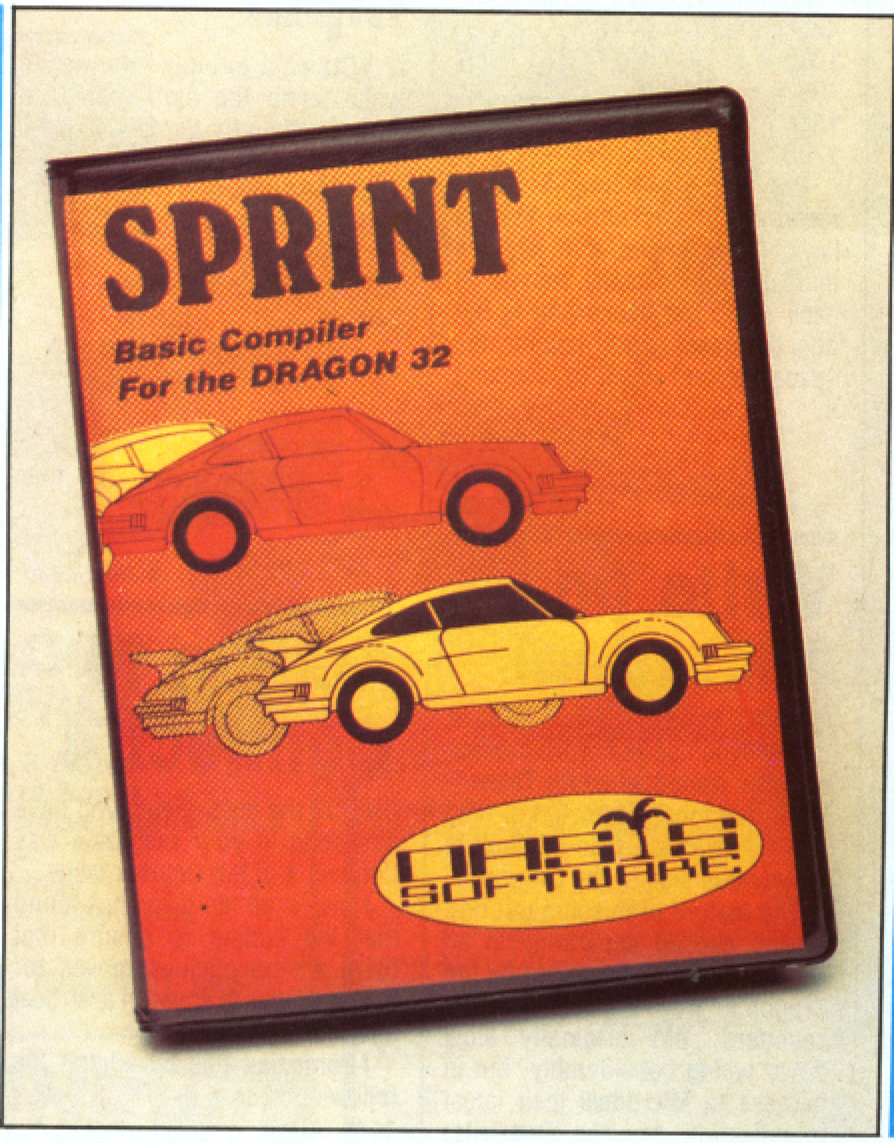
1	(2)	**Mined OUt**	Quicksilva
2	(7)	**Pettigrew's Diary**	Shards
3	(–)	**Night Flight**	Salamander
4	(5)	**Ring of Darkness**	Wintersoft
5	(9)	**Dragonfly Two**	Hewson Consultants
6	(–)	**Gridrunner**	Salamander
7	(6)	**Champion**	Peaksoft
8	(–)	**Lionheart**	Peaksoft
9	(1)	**Frogger**	Microdeal
10	(–)	**Morocco Grand Prix**	Microdeal

Chart compiled by Boots

OASIS SOFTWARE

Oasis Software present...

The first basic compiler for the DRAGON 32



Sprint compiles a subset of standard Dragon Basic which covers arrays, strings, four next loops, in fact virtually everything except floating point, arithmetic and associated commands. All arithmetic is integer and the Dragon sound and graphic commands are fully supported.

SPRINT BASIC COMPILER
by Dr. David Gray
FOR THE DRAGON 32

The Technique used is based on the approach used in U.C.S.D. Pascal where the Basic program is first reduced to intermediate code and this is then executed using a run time package which is saved with the rest of the compiled program.

- ★ Programs will run 5—10 times faster.
- ★ Almost the entire Basic is supported, with the exception of floating point commands.
- ★ Code produced will run independently of the compiler (for potential authors!)
- ★ Programs are compiled from tape under remote control so that much larger programs can be compiled.
- ★ Sprint is designed for ease of use and a comprehensive manual is included.
- ★ Free demonstration program with each program bought to illustrate the full power of the Compiler.
- ★ All Oasis products are covered by a lifetime Guarantee.

COMING SOON ....
DRAGON PASCAL £14.95

Dragon Pascal is an extended integer subset of the structured programmed language Pascal. A few of it's many features include:—

- ★ A complete set of structured programming constructs

  IF. .THEN. .ELSE WHILE. .DO CASE. .OF

- ★ COMPILER, EDITOR and SOURCE simultaneously resident for a rapid development cycle and total ease of use.
- ★ Very rapid compilation. Source can actually be compiled more rapidly than it can be listed!
- ★ Fully recursive.
- ★ Supplied complete with sample programs including routines which demonstrate techniques for simulating floating point functions such as SINE and COSINE.

OASIS SOFTWARE Alexandra Parade
Weston-super-Mare Avon BS23 1QT

Please send me

SPRINT Basic Compiler £14.95 ☐

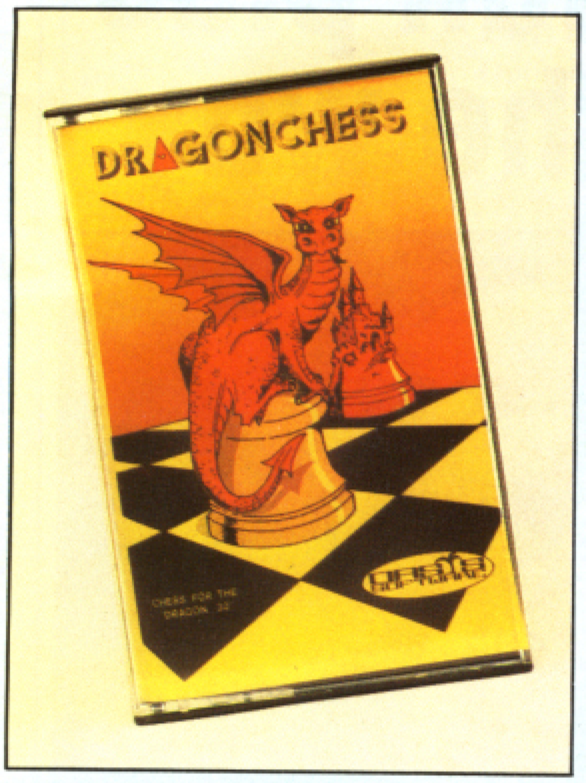
Name ....

Address ....

....

ACCESS orders
taken by phone
24 hours a day — 0934 419921

DRAGON 32 SOFTWARE...

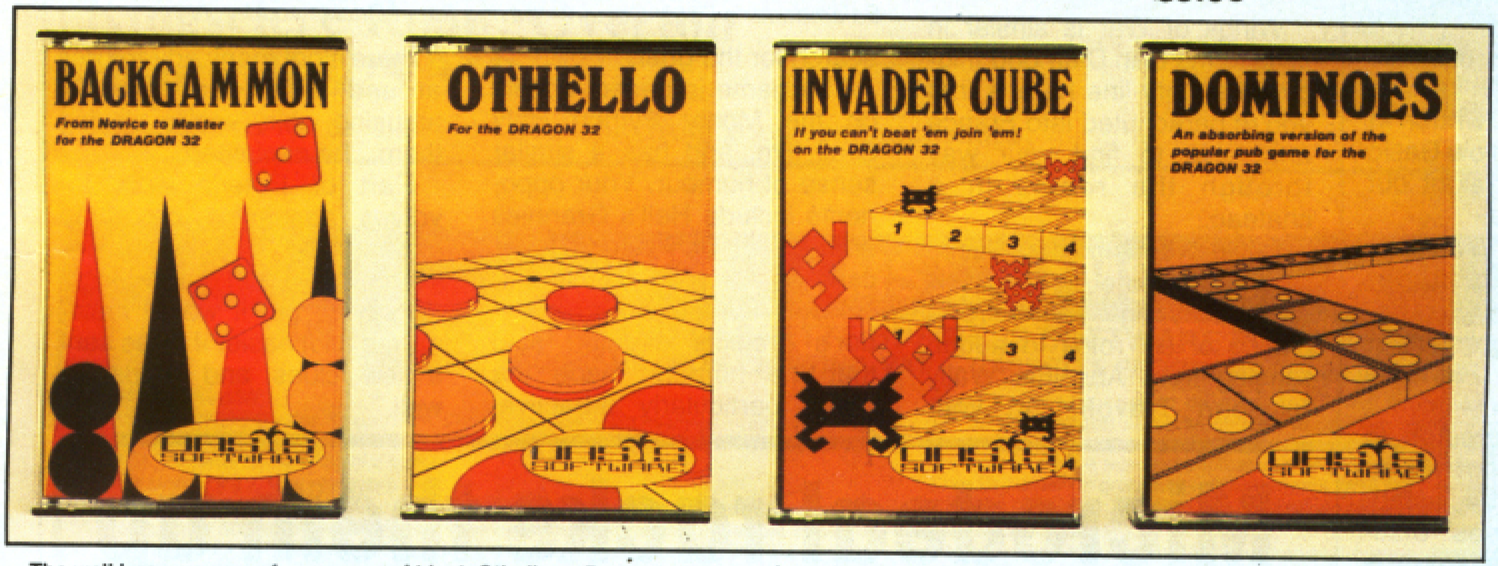


DRAGON CHESS £9.95

- ★ Six levels of play.
- ★ All legal chess moves including en-passant, castelling and pawn to queen promotion.
- ★ List of previous moves stored which can be displayed or printed.
- ★ Loading and saving of games positions from tape.
- ★ High resolution graphics which can be flipped round to make black or white play from either end.
- ★ Simultaneous text and graphics.
- ★ Slides can be exchanged at any stage.
- ★ Best move hint.
- ★ Set up from any position.
- ★ Opening move library.
- ★ Moves may be taken back and play resumed from any point
- ★ Change level of play at any point in the game.
- ★ Will adjudicate games between humans.
- ★ Very high standard of play.
- ★ Professional packaging and life-time guarantee.

Dragonchess is without doubt the best value chess on the market today. . . . MICRODEAL

BACKGAMMON £5.95 | **OTHELLO £5.95** | **INVADER CUBE £5.95** | **DOMINOES £5.95**



Backgammon: The well known game of backgammon complete with full instructions and computer demonstration for beginners.

Othello: At last. Othello or Reversi as it's sometimes known running on your Dragon 32. 2 levels of play, full instructions and computer demonstration for beginners.

Invader Cube: As well as being one of the best games of skill written for the Dragon 32 it also has some of the best pure machine code graphics we have ever seen.

Dominoes: Two games in one with full instructions. Hours of frustrating fun this program has 6 levels of play and on the higher levels adapts it's play to it's assessment of your ability.

Mind Games Compendium —— All five games for just £19.95

ALL OASIS PRICES INCLUDE VAT AND P&P AND EVERY OASIS PRODUCT IS PROMTLY DESPATCHED AND INCLUDES A LIFETIME GUARANTEE.

If your local dealer does not stock these products then let us know his address and we will contact him.

ACCESS ORDERS TAKEN BY PHONE 24 HOURS A DAY —— 0934 419921

Access

OASIS SOFTWARE ——
The Company that specialise in computer utilities.

OASIS SOFTWARE
ALEXANDRA PARADE
WESTON-SUPER-MARE
AVON BS23 1QT
TEL 0934 419921

Please send me

DRAGON CHESS	£9.95 ☐
BACKGAMMON	£5.95 ☐
OTHELLO	£5.95 ☐
DOMINOES	£5.95 ☐
INVADER CUBE	£5.95 ☐
MIND GAMES COMPENDIUM	£19.95 ☐

I enclose cheque/PO for £

NAME ....

ADDRESS ....

Shards additions

NEW games continue to emerge from Shards following its success with Pettigrew's Diary and Empire.

The company launched two games at the end of last year, Hooked and Monster Maths.

At £6.95 Monster Maths joins Shards' growing list of educational titles. It is a menu-driven mathematical cassette for 8-14-year-olds.

Hooked, on the other hand, is a fishing game presented in hi-res graphics which involves catching and landing as many fish as you can.

Cotswold comms

COTSWOLD Computers can now supply software linking modems to established databases and easing interface communications with the outside world.

Cotswold already markets an RS232 interface at £49.50. Adding the software, written in machine code, will allow Basic commands to go straight out through the interface.

The package, including documentation and tape, costs £10.00, and was developed by software engineer Tony Richards of Richards Systems.

Tony is also interested in writing modem software, enabling links to be established to databases elsewhere. One example quoted is of linking to the Eurolex legal database.

Tony can be contacted by writing to Cotswold Computers, 6 Middle Row, Chipping Norton, Oxfordshire.



Adventure fun plus turtles on the way

MORE adventures are on their way from Salamander — along with a Turtle graphics package.

Turtle Graphics costs £9.95 and will be released at the end of January. It has a comprehensive range of Turtle-type commands and a "huge manual".

You can design patterns, manipulate images, repeat shapes and achieve perspective effects by using string handling. The package is compatible with the Tandy four-colour printer/plotter.

The two new adventures, due at the same time, are Wings of War and The Cricklewood Incident. Each costs £7.95.

Wings of War is similar in style to the Dan Diamond trilogy. The story-line here is that you're parachuted into France and have to find your way through the rooms in a chateau.

Cricklewood Incident takes a more humorous approach to adventuring. You have a choice of six roles to adopt, ranging from Absolute Wally to John Travolta, in your search for the grail.


Salamander's Peter Ohlson – having fun in Cricklewood

The game is based loosely on the Monty Python Holy Grail film, and has a similar sense of humour. Your opponents include Hell's Grannies and a hail of Foster's lager cans.

Peter Ohlson, Salamander's projects director, said that the humour made the game particularly attractive. "There's not many adventures around which are actually amusing while you're playing them," he explained.

Part of Peter's work involves assessing programs sent in by Dragon users. "Some are worth developing," he says, "but it would make my life a lot easier if I was sent a few clues and maps as well."

Microdeal racks them up

MICRODEAL went into the New Year with more than 50 software titles under its belt, and a range of new releases are being planned for this Easter.

The most recent games from the company are all aimed at arcade fans — with the exception of two simulations, Pinball and Eight Ball (a version of Pool).

The arcade titles include Space Raiders, which is "a much, much better version of Space Invaders", and two games from US author Ken Kalish, whose past successes include Cuthbert in the Jungle and Phantom's Revenge.

New from him are the 3D game Danger Ranger, and Devil Assault which has three different screens and five levels of play.

Microdeal's list of UK-written programs is also increasing. Dave Thatcher, who wrote Cuthbert goes Walkabout, has contributed Dragon Hawk; Rick Redmen has written a real-time version of Star Trek called Space Fighter; and Skramble, with five different screens, comes from Steve Back, who wrote two earlier Cuthbert titles.

Each game costs £8.00. The price goes up to £19.95 for the more serious programs such as recent releases Rainbow Writer, Teleforth (which includes a tutorial and a Forth screen editor) and Filmastr (a strangely spelled title with a familiar application — databases).

Rainbow Writer is similar in concept to the word processing package Telewriter. It offers lower case, but this time for normal Basic. It also enables you to define your own alphabet for such things as foreign languages.

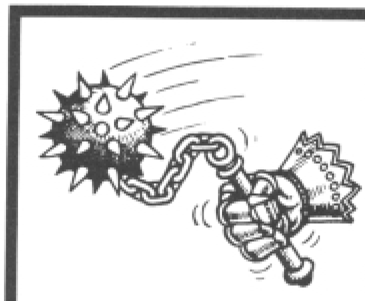
Microdeal's John Symes added that "a lot of other programs are on their way for the Dragon — probably towards Easter".

The company will have larger stands at this year's PCW and Earls Court shows enabling users to try out more games — "possibly 30 monitors at each".

Being worked on at the moment is a disk adventure with graphics, while a Cuthbert follow-up, Cuthbert in the Mines, is planned for February.

Microdeal is also starting a Cuthbert Club. Membership is free (entry forms are included with each game cassette) and entitles you to a quarterly newsletter featuring high scores, programming tips, etc.

WINDRUSH MICRO SYSTEMS



MACE by Graham Trott

**EDITOR
ASSEMBLER
MONITOR**

£29.95

DRAGON 32 CARTRIDGE

An 80 page reference manual describes MACE, the architecture of the MC6809 processor and its powerful addressing modes. The MC6809's 78 basic Assembly language mnemonics are also described in considerable detail.

The **EDITOR** can .... Insert line(s), delete line(s), overlay a line, replace a line, append text to a line, move up/down one line at a time, find a string, change string 1 to string 2, load a file from tape, save a file to tape, print (list) lines, call the ASSEMBLER or the MONITOR and return to BASIC.

The **ASSEMBLER** can .... Assemble without a listing to check for errors, assemble to screen, assemble to printer, assemble to tape, assemble to memory, assemble to memory with an offset, assemble with a sorted cross reference table and many combinations of these. The Assembler also provides TEXTUAL error messages to help locate programming mistakes quickly. When an error is encountered you can INSTANTLY return to the EDITOR and fix it! MACE supports local labels and global labels up to 8 characters in length.

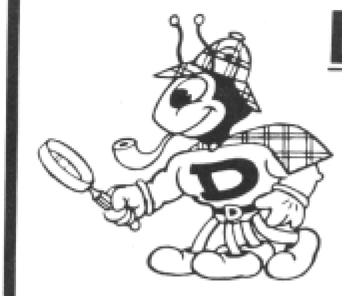
The **MONITOR** can .... View memory, examine and change memory, insert a character into a range of memory, shift the contents of memory from one location to another, find a byte or string of bytes in memory, set/clear a breakpoint, display the stacked registers, jump to a user program and continue execution of a user program after encountering a breakpoint. A simple hexadecimal calculator is also provided.

MACE's EDITOR/ASSEMBLER/MONITOR, DRAGON's BASIC, a BASIC program, an Assembly Language Source program, and machine code produced by MACE may all reside in memory together! You have INSTANT access to any of them.

R E V I E W S

PERSONAL COMPUTER NEWS ...... 'I would recommend MACE to anyone who wants to use the DRAGON to learn 6809 assembly language or to write serious assembly language programs'

PERSONAL COMPUTING TODAY .... 'This package has a very professional feel about it ...'



D-BUG by Stewart Bush

**TRACER
MONITOR
DISASSEMBLER**

£29.95

DRAGON 32 CARTRIDGE

D-BUG is a complete software emulation of the MC6809. This enables D-BUG to maintain complete control of the real MC6809 when illegal instructions or program errors are encountered. Properly used D-BUG will make working with assembly language as easy as BASIC but a lot more rewarding!

D-BUG is **POWERFUL** It is a program analysis tool that combines a single step tracer, a disassembler, and a system monitor all in one. It even has its own RAM so it does not need to steal one single byte of the DRAGONs 32K!

D-BUG is **FRIENDLY** It's 36 commands are instantly available through a menu to simplify use and eliminate systen crashes caused by errors.

D-BUG is **THOROUGH** During program trace it follows the execution of program code an instruction at a time under your full control. As each instruction is executed the contents of the 6809 registers are displayed along with a dissassembly of the instruction about to be executed.

D-BUG is **COMPLETE** Everything you need to analyze and de-bug a machine code program resident anywhere in the DRAGONs 32K memory is provided in one ROM/RAM cartridge. D-BUG does not use the SWI instruction for its operation so it is capable of working with program code in ROM. This **UNIQUE** feature enables you to trace through the BASIC ROM code and see what makes it tick!

D-BUG is **IMPOSSIBLE** to describe fully in this limited space so we will just give you a summary of the available commands: Single step trace; auto-trace; execute current subroutine; disassemble; set breakpoint; set subroutine nest level counter; alter target 'CC', 'A', 'B', 'DP', 'X', 'Y', 'U', 'SP' or 'PC' register; display target registers; memory examine/change; byte finder; fill memory; view memory in HEX and ASCII; calculate relative branch offset; shift blocks of memory (they may overlap); jump to subroutine; write title/notes to printer. Toggle (on/off) functions include: echo output to printer; execute BASIC ROM routines at full processor speed; trace/disassemble; dump registers after each command; display registers after each instruction when in auto trace; delay between instructions when in auto-trace.

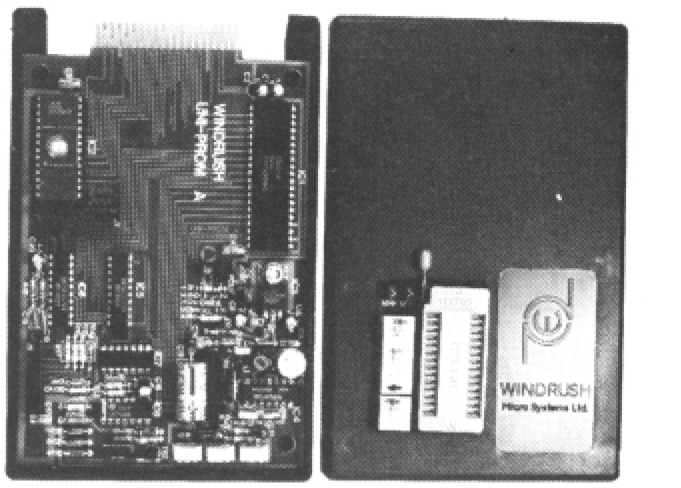


BUG ZAPPER

MULTI-PROGRAMMER

£79.95

DRAGON 32 CARTRIDGE



PROFESSIONAL QUALITY EPROM PROGRAMMER FOR THE DRAGON-32

Built-in ROM provides full range of menu driven EPROM programming facilities including: FILL, MOVE, EXAMINE/CHANGE, HEX-ASCII DUMP, CRC CHECKSUM, COPY EPROM, VERIFY EPROM, PROGRAM EPROM, START/STOP CASSETTE MOTOR, SAVE DATA TO CASSETTE and LOAD DATA FROM CASSETTE.

EPROMS supported include: Intel 2758, Texas 2508, Intel 2716, Texas 2516, Intel 2732 and 2732A, Texas 2532, Motorola MCM68764/6, Intel 2764*, Texas 2564, Intel 27128*, and Intel 27256*. TRI-VOLT EPROMS are NOT supported!

* Intels 'Intelligent Programming Algorithm' (tm) is available for these devices. This algorithm enables these devices to be programmed in one sixth of the conventional time with a dramatic improvement in programming margin.

6809 FLEX(TM) SOFTWARE/HARDWARE

PL/9 by G. Trott	EDITOR/COMPILER/TRACER	114.00
MACE "	EDITOR/ASSEMBLER w/XMACE EDITOR/6800/3 XASSEMBLER	57.00
ASM05 "	EDITOR/6805 CROSS ASSEMBLER	57.00
GT-BUG "	6809 SYSTEM MONITOR (with source listing)	30.00
'C' by J. McCosh	UNIX VII COMPATIBLE ... W/FLOATS & DOUBLES	191.00
D-BUG by S. Bush	6809 TRACER/DISASSEMBLER	57.00
INTROL 'C'	INTROL 'C' COMPILER	547.00
OMEGA PASCAL	OMEGASOFT PASCAL COMPILER	375.00
OMEGA RALL1	OMEGASOFT RELOCATING ASSEMBLER/LINKER	104.00
LUCIDATA P69-3	LUCIDATA PASCAL	145.00
SCREDITOR III	EDITOR/WORD PROCESSOR	145.00
DYNACALC	ELECTRONIC SPREAD SHEET	175.00
DYNAMITE+	DISASSEMBLER	175.00
RMS	DATA BASE MANAGER	206.00
EDITOR	TSC TEXT EDITOR (SP09-02)	61.00
ASSEMBLER	TSC 6809 ASSEMBLER (SP09-03)	61.00
DEBUG	TSC DEBUG PACKAGE (SP09-05)	84.00
UTILITIES	TSC UTILITIES PACKAGE (SP09-11)	84.00
DIAGNOSTICS	TSC DIAGNOSTICS PACKAGE (SP09-12)	84.00
PASCAL	TSC NATIVE CODE PASCAL (SP09-15)	191.00
XBASIC	TSC EXTENDED BASIC W/17 DIGIT MATHS (SP09-06)	107.00
XBASIC-PC	TSC EXTENDED BASIC PRE-COMPILER (SP09-08)	61.00
SORT/MERGE	TSC SORT/MERGE PACKAGE (SP09-10)	107.00
JCP	JOB CONTROL PROGRAM (AN ENHANCED 'EXEC' FACILITY)	73.00
FLEX TOOLKIT 1	FLEX DISK REPAIR PACKAGE (FRANK HOGG)	74.00
FLEX TOOLKIT 2	EXTRA FLEX UTILITIES (FRANK HOGG)	74.00
BASIC TOOLKIT	EDIT/DECOMPILE/CROSS-REFERENCE BASIC (FRANK HOGG)	74.00

WE ALSO SELL PROFESSIONAL MC6809 DEVELOPMENT SYSTEMS RUNNING FLEX AND OS-9

SYSTEMS AVAILABLE IN DESK-TOP, SS-50, AND 3U CONFIGURATIONS

FLEX is a trademark of Technical Systems Consultants
OS-9 is a trademark of Microware Systems Corporation
DRAGON is a trademark of Dragon Data Limited

WE ARE DEALERS FOR: TSC, MICROWARE, GIMIX, SSB, FRANK HOGG LABS, AND LUCIDATA

**PRICES INCLUDE POSTAGE & VAT.
VISA/ACCESS
PHONE ORDERS ACCEPTED**

**WORSTEAD LABORATORIES,
NORTH WALSHAM, NORFOLK,
ENGLAND. NR28 9SA.
TEL: (0692) 405600**

Welcome to Dragon World

THE *Stop Press* newsletter from Dragon Data has grown in size and changed its name to *Dragon World.*

The first issue came out in December, taking over from issue 5 of the newsletter. Subsequent issues will be published once every two months.

There's 16 pages in December's *Dragon World,* and Dragon Data expects the February issue to be bigger.

The magazine is mailed out free to users who have returned their warranty cards.

The address for contacting the new magazine is *Dragon World,* Dragon Data, Kenfig Industrial Estate, Margam, Port Talbot, West Glamorgan.

The contents of the first issue are similar to *Stop Press's* — with a machine code corner, young user pages, readers' letters and a selection of programs.

Additional features are a technical advice column and user club news.

ITL ponders 3 inch Dragon disk drives

ITL HOPES to produce an interface for its Byte Drive 500 three inch disk drive which will make it compatible with the Dragon, despite problems with one of the Dragon's interface chips which may necessitate a retro-fit.

The 3 inch disk format is said to offer more "bytes per pound" than the more common 5¼ inch type, and may well become the small business market leader when a full range of applications software is available.

ITL's Tom Boyle commented that the potential of the Dragon was such that it would be unfortunate if the technical problems involved could not be overcome.

Then the Byte Drive 500 could compete with the two Dragon disk drives already available, from Dragon Data and Cumana. Tom expected to see three inch drives with one megabyte capacity being available.

Work on the cable interface for the Dragon, which should cost around £100 with the disk drive and manual, should begin once ITL has finished developing Sinclair Spectrum and Commodore 64 versions.

A full range of software, including assembler/disassemblers, text editors, spreadsheets and databases, is now in preparation. ITL also hopes to make several popular games available on three inch disk.

Pick a printer from Tandy



Tandy's CGP-220 – seven colours for £149

MORE THAN 10 Dragon-compatible printers are now available from Tandy — ranging in price from under £100 to over £1,000.

The TP-10 Thermal Printer is the cheapest, at £79.95. It prints at 30 characters a second on 4⅛inch wide thermal paper which costs £2.99 a roll. This is the only printer in the range requiring special paper.

The CGP-115 four-colour printer plotter at £149 is already well-known. More recent is the CGP-220 which adds another three colours (yellow, violet and magenta) and increases the print speed to 40 characters a second in text mode.

These printers are said to be ideal for line listings. Worth considering for low end letter quality printing is the DMP-110 at £299. This is a 9 inch dot matrix printer with word and data processing modes.

Print speed is 50 characters a second — or 25 for word processing. For better quality look at the £599 DWP-210 which prints at 18 characters a second.

Higher up the range is the £699 DMP-420 which Tandy describes as "excellent for the small business user with big throughput". The company expects the market for such printers to rise as the OS-9 operating system takes off.

Fanfold paper up to 15 inches wide can be used with this dot matrix model which prints at 140 characters a second.

Top of the range is a daisy-wheel model at £1,399.

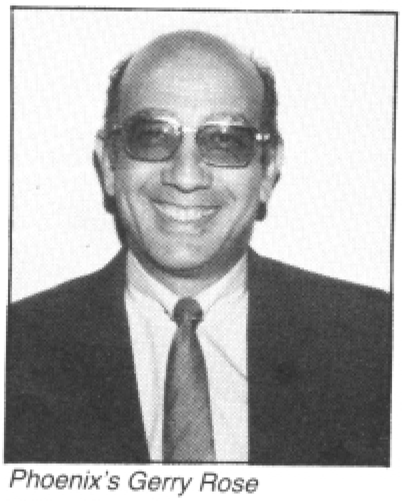
First games for the 64

PHOENIX Software is one of the first companies to take advantage of the Dragon 64, offering a package aimed at adventurers and arcade fans alike.

The Emperor Must Die is a twin-cassette package released this month at £9.99 — one cassette features an adventure and the other arcade action.

The story line is that the emperor of the galaxy is corrupt and you are chosen to assassinate him, assuming that you can pass the tests set along the way.

These tests present themselves in the form of three "trips" to the action cassette (using 22½K of memory) and two to the adventure plus one brief visit (using 48K).



Phoenix's Gerry Rose

The Emperor Must Die is Phoenix's first offering for the 64. Previous software from the company, set up last year by Gerry Rose, also featured the twin-cassette concept, but was for the 32 only.

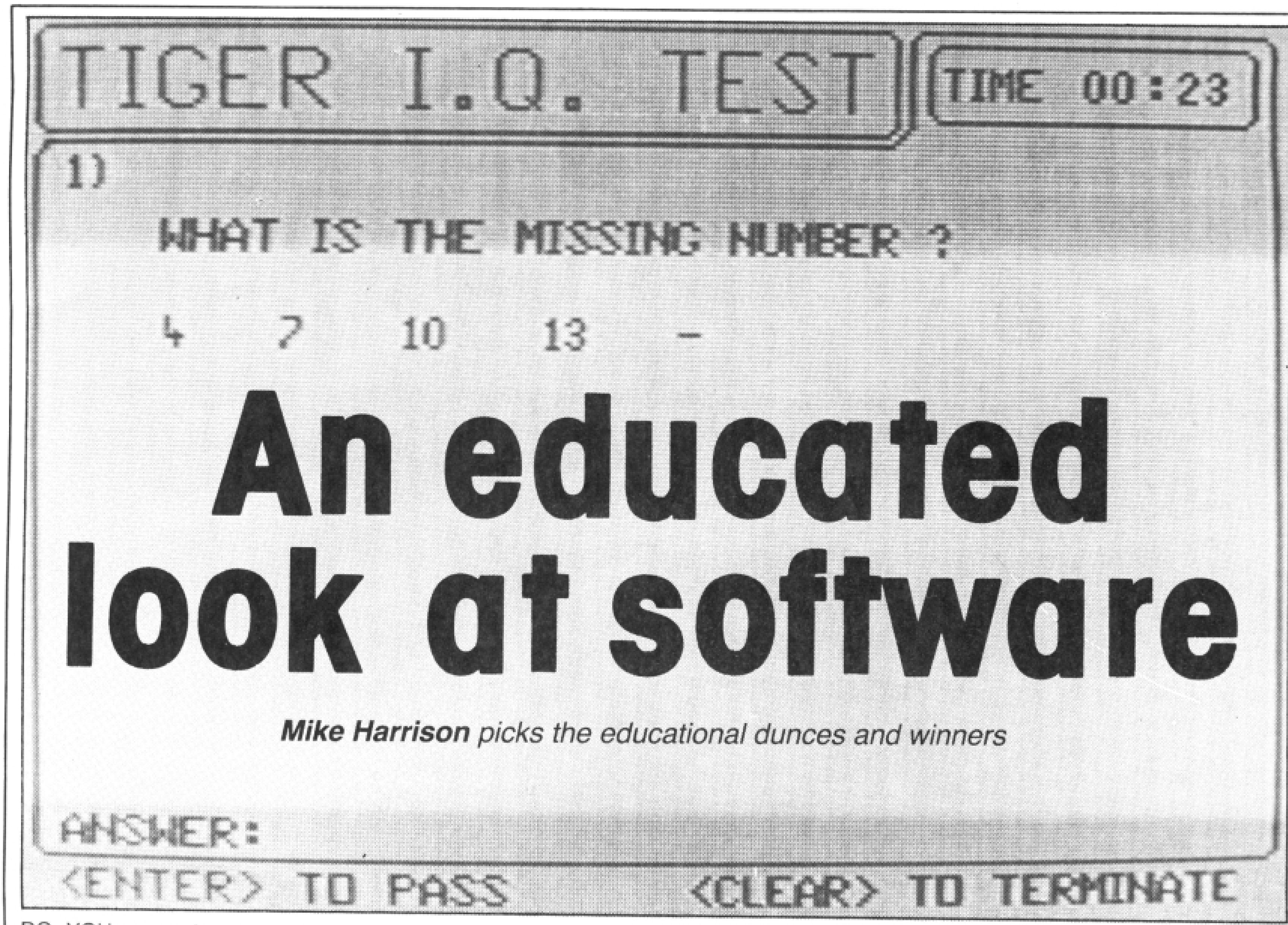
You begin with the action cassette, a space game where you have to reach a randomly arranged sequence of control towers on different planets.

Reaching a third tower gives you the running code for the first adventure, set on an unknown planet where you have to survive against the native inhabitants, wild animals and an unfriendly terrain.

Surviving gives you the locations of the first five control towers on the action tape and the correct order for reaching them.

Doing this successfully gives you the running code for the second side of the adventure — in which you recruit personnel and buy equipment to build a rocket capable of defeating the emperor.

Success here takes you back to the second five control towers on the action tape, collecting clues along the way for the third mini-adventure. The coup d'etat should then follow on the eleventh planet with your assassination attempt proving successful.



An educated look at software

***Mike Harrison** picks the educational dunces and winners*

DO YOU remember those heady days when you first got your Dragon? How it was going to amuse, enthuse and educate your family. How your finances and home management problems would be a thing of the past and children's learning be smoothed.

Well, think on. Just how many distressed maidens have you rescued, frogs have you squashed and Klingons zapped? Compare this to the educational use your computer has had. Up to now you've had the excuse of lack of appropriate software. Is that true now? Every advert seems to slip in an educational tag so let's see if the excuse still holds up.

Spelling

Talking Speller, for example, is an ideal program to encourage children to learn those school spelling lists. You know, the ones they produce from their pockets for the first time over breakfast on the day of the test itself. Schools all seem to set these tasks but seldom advise on how best they should be learnt. Help in the shape of this £9.95 Eurosoft tape is at hand.

The user creates a data file, with careful input controls to ensure a good chance that words on the file are in fact spelled correctly and then helps the child make a voice-track on tape for each of the words.

Children can then test themselves using these audio cues in their own accent, at their own speed either immediately or at some time later by reloading this data. The strength of this program is also shown in error handling. It tells the user if his response is too short or contains too many letters and shows children the letters they've got in the right places so they can immediately make some attempt at correction.

The novelty value of the tape as it stands is a strong incentive to learning. The added graphics drawing capability in the version now on offer makes this a good educational program. It is not of course limited to learning spelling lists. I used it as a French/English vocabulary primer and it could be used in any circumstances calling for the Dragon's special ability to transmit sounds from tape to TV.

The Dragon 32 missed out on the Department of Industry micros in schools scheme, possibly because its text is only in capitals. Children's reading is always in lower case and although it is possible to draw these in high resolution many educational programmers have missed this point, even when producing material for infants. It is ironic therefore that Galactic Hangman which is played entirely in hi-res still uses capitals drawn on the screen. This cassette is also from Eurosoft and costs £7.95.

The unfortunate prisoner is saved from hanging if you can guess the word. In fact a spaceship destroys the whole jail in this event but what happens to the other inmates we never do find out. Quite good fun and in Basic so the data can be accessed, but it is scarcely educational. It doesn't teach anyone anything. It doesn't reinforce spelling or even teach tactics or strategy because the language is so full of exceptions to contradictory rules. It has no role in developing language skills, as words need to be taught in the context of their meaning.

Silly Syntax, the third in the Eurosoft series, does do just that job. In Silly Syntax the importance of words is highlighted by the creation of funny stories much on the lines of Consequences, the party game. There are a number of basic storylines with players being asked to provide a plural, or adjective or nonsense word which is then injected into an appropriate part of the story to sometimes hilarious effect.

Many primary school children may have come across procedure exercises in school where every seventh word or so in a story is blanked out and from the context he has to supply an appropriate word. "Both Sally and John like .... chocolate", might encourage replies like milk, hot, Swiss or eating. The basic story can be shown with the "gaps" to be filled in Silly Syntax and there is a creative mode where children can enter their own work which has been stimulated by the game. All options can be output to a printer. Silly Syntax is £9.95 and 60 further stories ranging from Fairy-tales to X-rated (for adults only) are available.

Skills

Eurosoft's range of educational products also include Alps at £14.95 and Melody Express at £7.95. Alps is a Cesil compatible interpreter which allows easy entry into assembly language programming ▶

DRAGON 32 OWNERS

DELTA DISK SYSTEM

Make your Dragon turn into a real computer with the new Double-Density Delta Disk System.

The Delta Disk System Gives You . . .

- An affordable disk system.
- Powerful Delta disk commands.
- Lets you produce and handle random access files as easily as serial files.
- Random sequential and indexed file handling.
- Simple plug into Dragon there are no HARDWARE MODS needed to run DELTA!
- Easily expandable 180K to 1.4 megabyte ON LINE storage.
- Full range of business utility and games software AVAILABLE NOW!
- The price you see is the price you pay. NO HIDDEN RAM upgrade costs.
- Uses under 2K of user – RAM as DELTA is held in EPROM.
- Enables programmer to easily produce applications Software which automatically starts up and operates without any intervention from the user.

FULL RANGE OF BUSINESS SOFTWARE AVAILABLE. SEND SAE FOR DETAILS.

DELTA CARTRIDGE – contains DELTA disk Operating System, User Manual, demonstration diskette	£120.00
DELTA 1 – DELTA Cartridge, User Manual, a single-sided 40-track (180K) drive plus cable	£320.00
DELTA 2 – as DELTA 1, but with a single-sided 80-track (360K) drive	£355.00
Disk interface cable (supplied with DELTA 1 or 2)	£9.95
ENCODER 09 assembler/disassembler/editor – integral with DELTA	£24.95
HOME ACCOUNTS full home package for DELTA	£14.95
INFORM – Data Base Management System commissioned especially for DELTA Systems	£39.95

NEW

TOOLKIT FOR DRAGON 32

PREMIER'S ASTONISHING NEW TOOLKIT FOR THE DRAGON 32 PERFORMS THE FOLLOWING AMAZING FUNCTIONS:–

- FULL screen editor allowing copying, concatenating of lines etc. Fully linked to the DRAGON'S line editor.
- Eight PROGRAMMABLE KEYS – all easily accessed or checked.
- 25 full colour low-resolution graphics screens available with commands for instantly writing to any of them or swopping/moving. MAKES COLOUR ANIMATION AT MACHINE CODE SPEED a simple task.
- Full range of ERROR HANDLING commands with several RESUME options.
- OVER SIXTY NEW WORDS fully linked to your DRAGON BASIC – your new commands become built-in as soon as you plug in the TOOLKIT cartridge.
- VARIABLE GOTO/GOSUB commands, allowing routines to be called by name. Eg GOSUB TITLE or GOTO FINISH.
- Superb TRACE command which uses top right of screen only, thus leaving graphics/text intact.
- SEARCH and REPLACE commands for easier program modification.
- Comes complete with COMPREHENSIVE 40+ page MANUAL.

Available in CARTRIDGE or DELTA enhancement format .£29.95inc.

NEW

● DRAGON Fig FORTH ON DISK FOR DELTA ● Executes up to 10 times faster than BASIC ● Language extendable by user defined words ● Words can even be defined using the FORTH assembler for maximum speed ● BASIC and DELTA commands still available from FORTH ● Source code stored and can be compiled from disk ● FORTH can be saved to disk, including any new definitions ONLY £29.95 including MANUAL + Fig FORTH glossary + Model. Inc. VAT

SCRIBE FOR THE DRAGON 32

- FULL UPPER and LOWER CASE direct from the keyboard.
- An enlarged 42 x 24 screen display which gives a superb READABLE text.
- Full text display on hi-res screen.
- Automatic underline and reverse image.
- Up to 255 user-defined graphics.
- Precision character position command giving super/subscripts.
- Black on white text display option.
- A new Print @ command extending to 1008 thus almost doubling screen area.

ENCODER 09 FOR THE DRAGON 32

ENCODER 09 – is a full symbolic assembler using standard mnemonics and pseudo op-codes. Source code can be incorporated into BASIC programs. The monitor section contains commands to allow memory display, modification and execution. Memory block move, breakpoint handling, full disassembly and a full editor are only a few of its many features. The most powerful assembler/disassembler/editor available for the DRAGON 32. Available as either an integral DELTA fitment or in cartridge.
Cartridge £29.95. Disk – see above.

HIPRINT FOR THE DRAGON 32

HIPRINT – screen dumper.

– will dump the entire contents of your DRAGON 32 high-res screen to a high resolution printer. Can be used for design, display etc. Available at present for EPSON printers only. Other modules to follow shortly.
Tape £7.95. DELTA disk £9.95.

PREMIER MICROSYSTEMS

PREMIER MICROSYSTEMS
208 Croydon Road, Anerley, London SE20 7YX
Telephone 01-659 7131 or 778 1706
Dealer and export orders welcomed.

POSTAGE and PACKING
Tapes 1–2 95p, 3+ £1.25.
Cartridges £2.50
Drives or Disk packages £4.50
ALL PRICES INCLUDE VAT.
Send SAE for full catalogue.
ALLOW 28 DAYS FOR DELIVERY.

Access BARCLAYCARD VISA

◀ for 'O' level computer science. Melody Express provides a simple introduction to keyboard skills as an introductory stage to musical programming.

Tiger Software is another company which advertises "Educational Software — designed by experienced teachers". The Tiger IQ Test is beautifully drawn on the hi-res screen and looks for all the world like the exam paper it's meant to be. The test is timed and (a nice touch this) the clock only activated when the full page is complete and ready for your answer. There is a good variety of questions such as:

WHAT IS THE NEXT NUMBER IN THE SERIES
4 7 11 16 22 —
WHICH WORD IS THE SUFFIX
S
PL
BL —(— — —)
F

as well as special questions giving an all round test of this sort of reasoning (for the hard-of-thinking the answers are 29 and EAT).

The £7.95 tape contains two 40 question tests. At the end of the test your supposed IQ and an indication of your intellectual worth are announced. If taken as a bit of fun, or even as a practice to give you an idea of what to expect in IQ tests (more commonly called verbal reasoning tests), this program is fine but expensive. After all you can get paperbacks at £1.50 with dozens of tests in. Where I take issue with Tiger is in its use of the "Educational" flag of convenience — for this it certainly is not. An intelligence quotient is calculated with reference to a student's age. No request for age is made. I refuse to believe that a 10-year-old and 30-year-old getting the same score on this test have the same IQ. Tiger's test says they do. In fact we are not told if this test is for primary children, 15 year olds or tired executives (all properly constructed tests should have a target age range).

If you expect to get better at these tests by cribbing from the answer page you can forget that too. For although you can compare your list of answers to those of Tiger's this is in isolation from the actual questions which you cannot recall except by taking the test again. Anyway without knowing the reason for the "correct" answer no learning can take place at all.

Child-proofing

My final criticism of this tape applies to many others too, and concerns child-proofing. This means helping the user to show his knowledge and not make mistakes due to the computer's method of working. For example in the question

WHAT IS THE NEXT LETTER
A C E G —
a) H b) I c) J

some children typed in the letter 'I' which is correct reasoning but not the answer B which was acceptable. It is easy to restrict returns on a computer — so why not do it.

Child-proofing was also lacking on the £7.95 Tiger Grand Prix which is a racing game for one or two players. The players are assigned cars which go around a circuit by moves dependent on the throw of a dice and the answering of a general knowledge question. There are around 500 different questions in 13 data files suitable for five different age ranges from seven to adult. Younger players get questions including simple maths and spelling, and adults need to know obscure items from books of records. As in the IQ test no attempt is made to teach anything and although general knowledge quizzes may be fun, to sell them under the guise of "education" is both misleading and potentially harmful to this market.

Not much better is the £9.95 Eduquiz 1 from Gem Software. This takes the form of the TV quiz Winner Takes All, so it does have the value of allowing players to back their judgement by the size of their bets. It is well child-proofed but with questions like "Who was married to the Monarch whose reign began in 1422?" we have perhaps gone beyond the age where this is necessary. At a time when children are learning to break state security codes and pull down information from thousands of miles away, it seems incredible that these programs ask obscure and useless facts about mountain ranges, long dead rulers and 100-year-old inventions. This program even has a mistake. Islambad does not exist (the town in Pakistan is Islamabad), and San Cristobel is not the capital of Cuba either.

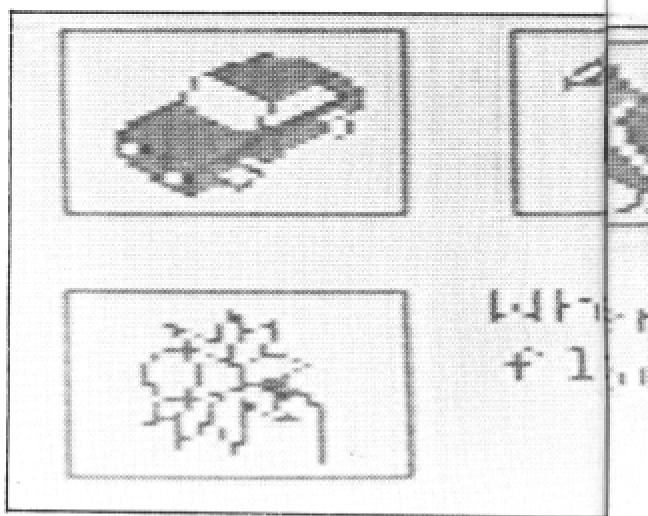
General knowledge

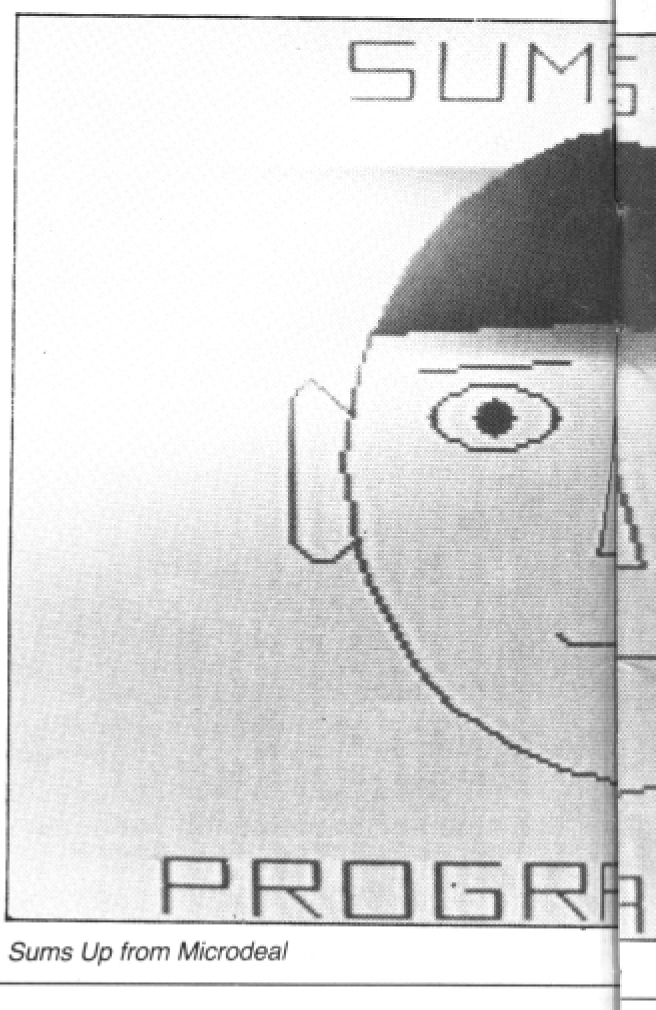
Perhaps the computer, then, is not the best way of testing general knowledge as misunderstandings cannot be dealt with and multiple choice questions are open to guesswork. The type of skill that the Dragon 32 is ideal for, however, has been exploited very nicely by Gravesend Home Computers, in its Teach Type. This £6.95 program aims to have you touch typing in 10 hours and shows the correct finger for each letter as it is introduced. Visual representation of accuracy and speed encourage you to look at the screen, not the keys as you type. Typing is such a useful skill for teenagers to have in the fields of computers, journalism, further education and clerical work that Gravesend deserves to succeed with this program. Those who are learning already may find the absence of home keys a little daunting but will be relieved from ASDF;LKJ boredom. The Dragon keyboard being one of its stronger features over its rivals, Teach Type ought to be a winner.

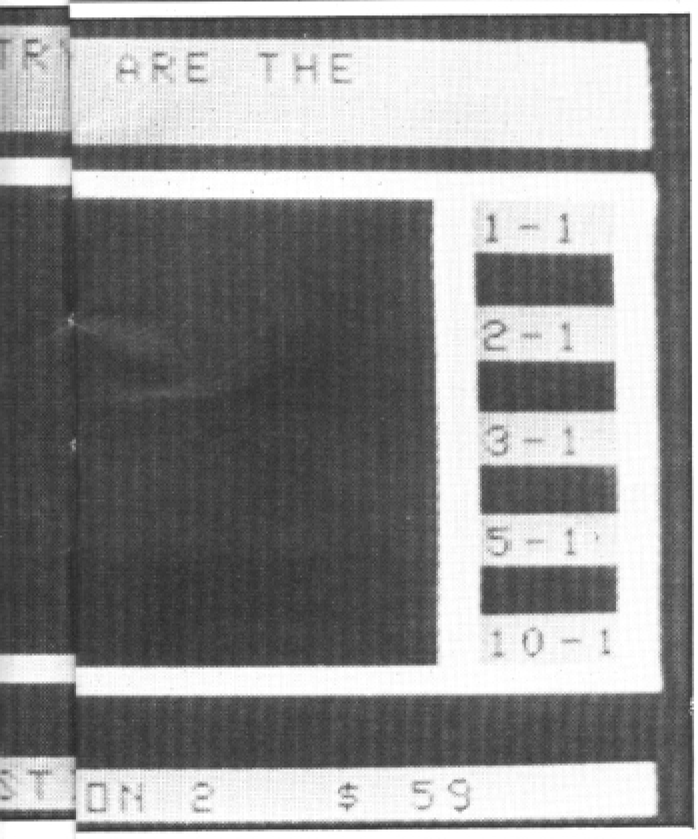
Tick Tock is a courageous attempt by the same company to design a program suitable to help reinforce time-telling skills in young children. A friendly clock (looking not a little unlike Ivor the Engine's face) is drawn on the screen, and hours, half-hours and quarter-hours are displayed for children to read. The face rewards you with a wink and a smile when right. The display is colourful, chunky and appealing

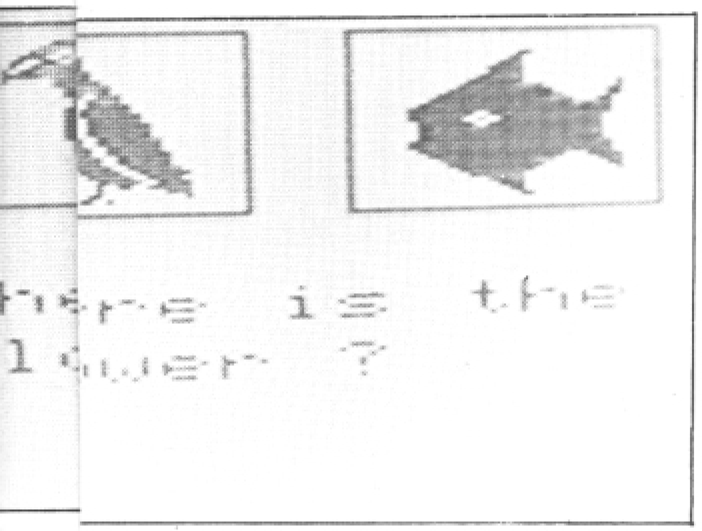


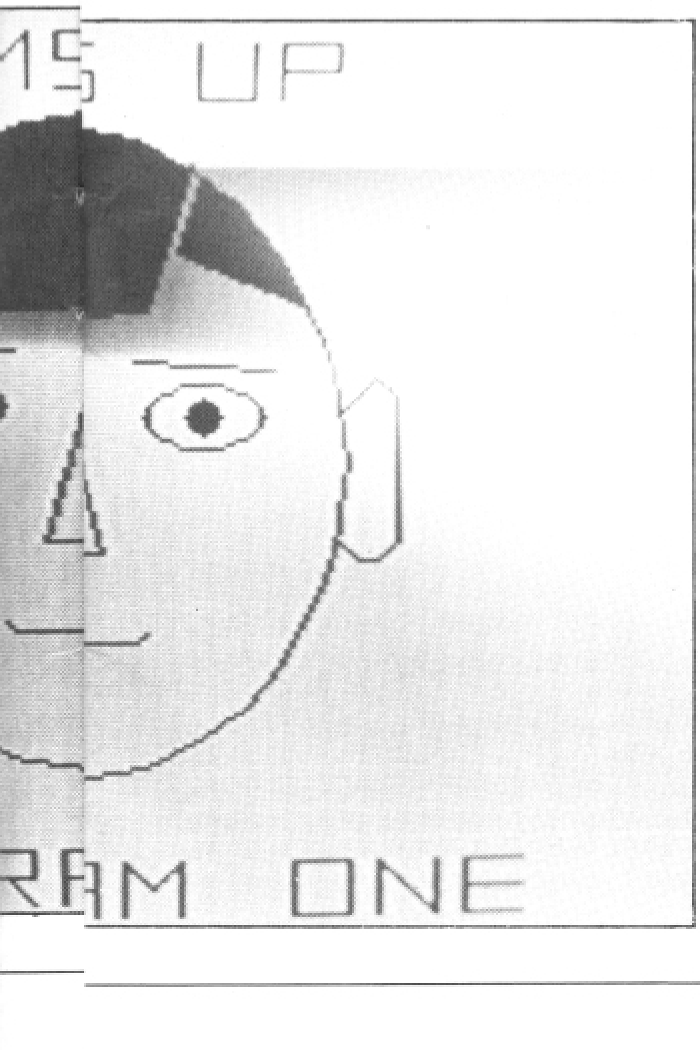
Gem Software's Eduquiz Geography


Dragon Data's Hide and Seek


Sums Up from Microdeal







to children and the hickory dickory dock theme attractive.

The program has however, a few flaws. The key to successful time-telling is to distinguish between the long and short hands and to read the figures clearly. Making the hands different colours is not good enough (some people use monochrome TVs) and real clocks have hands the same colour. The eyes and mouth on the clock face are confusing and distract attention from the real action. One eye makes 10 look like 18. When the child has made a mistake it is not corrected and once a key has been hit it cannot be backspaced. The method of answering requires quarter past two to be entered by 2, colon, 1, 5 or it's wrong.

It is quite possible, if the real purpose of the tape is to teach time, for these difficulties to be overcome by re-programming, but essentially what is needed is an experimental phase to precede these tests. Why not allow children to put in some times and then make the clock show them? Tests could then be based on what the child himself had keyed in previously. Lastly children up to six or seven have not learnt 6.45 or 5.30 but still talk about quarter-to and half-past. If revamped this £5.95 program could possibly be the finest of the bunch, and would be in demand in infant classrooms as well as at home.

Drowning

Baby Dragon (£6.95 from Gravesend) contains two programs for young children. Koko does nothing more than any Dragon user could, generating random numbers to be multiplied, divided, added and subtracted. All the action takes place on the text screen and no one has bothered to child-proof it. Letters are bound to be entered by mistake by young children and the consequent REDO? wipes out part of the print (a) graphics. The program has to be run again if this happens or if break or clear are touched. The object of the exercise seems to be to drown poor Koko (not very sporting) but when this happens the water rises up against gravity to cover him. This is, of course, graphically easier than having him fall in but difficult to explain to an enquiring six-year-old.

Much better in presentation is the other program Teddy. A number of honey pots seek your help in bouncing past sleeping bears who in turn hope to catch some, and suddenly sit up to do so. The hi-res graphics are very well drawn and the game is lots of fun for little children who only have to control the jumping by use of the space bar. I would say that three and four year olds would learn quite a bit trying to decide when to jump and if a score of successful escapes were displayed it would be a learning situation.

Unfortunately no scores are displayed for the program is yet another test. The children are supposed to count the honey pots out and count them back in again (selection for BBC war correspondents?). However they won't be able to read the (capitals only) questions nor to remember to keep adding on the pots to their running total to satisfy the examiner. So I'm afraid it's thumbs down for Baby Dragon.

Many people who have recently purchased their Dragon from a large department store will be familiar with Ampalsoft's Cheshire Cat Basic Tutorial. A package for younger children in the same series is Maths 1. It is a very versatile program. Options on difficulty level, display of players' scores, changing the running order of exercises and the number of different questions per exercise all exist. Each of the 10 different exercises is introduced by a nursery jingle and difficult reading is avoided as the programmers assume that an adult is around to help out where necessary.

Kangaroos

The players' names (drawn on hi-res screen but unfortunately in capitals) act as a prompt for their answers. Only numerical answers are accepted and wrong answers are carefully shown to be so. The program is geared to learning rather than just testing and is beautifully designed. Children are asked to count the balls kicked into a tube (of Fosters?) by a kangaroo, and asked to make numbers of boats float into harbour in response to numbers shown. Sorting, mapping, sets and simple addition are included in these exercises, including a lovely one towards the end where children have to find the tallest and shortest in a line of flowers.

The package comes with two complete tapes (both double recorded) in a large plastic folder with some documentation. Unfortunately, for security reasons, the programs load additional data when running so breaking the program means the tedious process of reloading. However, Maths 1 will certainly give young children a good start in practising simple numbers before they start school and despite its £19.95 price it is good value for money.

Older children need practice in tables and numbers too and Allyn Software has two £7.95 programs to provide it. Sums 1 starts with a menu option for the four rules of number and a comprehensive set of instructions. The sums are presented in hi-res and there is graphic representation of score and time left. The slowness of the Basic means that keyboard responses against the clock become difficult. To enter 44 the sequence 4, 4, <ENTER> is too fast and is registered as 4 <ENTER> and marked wrong. Once you slow down and get the hang of it you can overcome this.

By the time Allyn Software produced Sums 2 it had learned the lesson of lower case and presents seven pages of instructions in a style most children will be able to read. This is ironic as the subject matter — manipulation of fractions — is suitable only for children five years older than those for whom Sums 1 might be useful. Wrong answers are erased and the method for gaining the right one shown. Perhaps future programmers would bear in mind that the computer lends itself easily to showing addition and subtractions of fractions in pictures (of cake for example), and incorporate this in their programs.

However, it is a puzzle to me why anyone wants children to clog up ▶

◀ their minds with ways for finding ⅞ of 2⅓, especially when we have machines as cheap as calculators to do it for us. No one expects us to do without our lawnmowers and learn to cut the grass by hand, or to learn how to rub two sticks together to roast the Sunday joint. I've been perfectly able to manipulate fractions for 25 years but still am waiting for it to be put to some use. However, some schools do still require pupils to learn these things and if your son or daughter is having difficulty then maybe Sums 2 might be the answer.

Circus

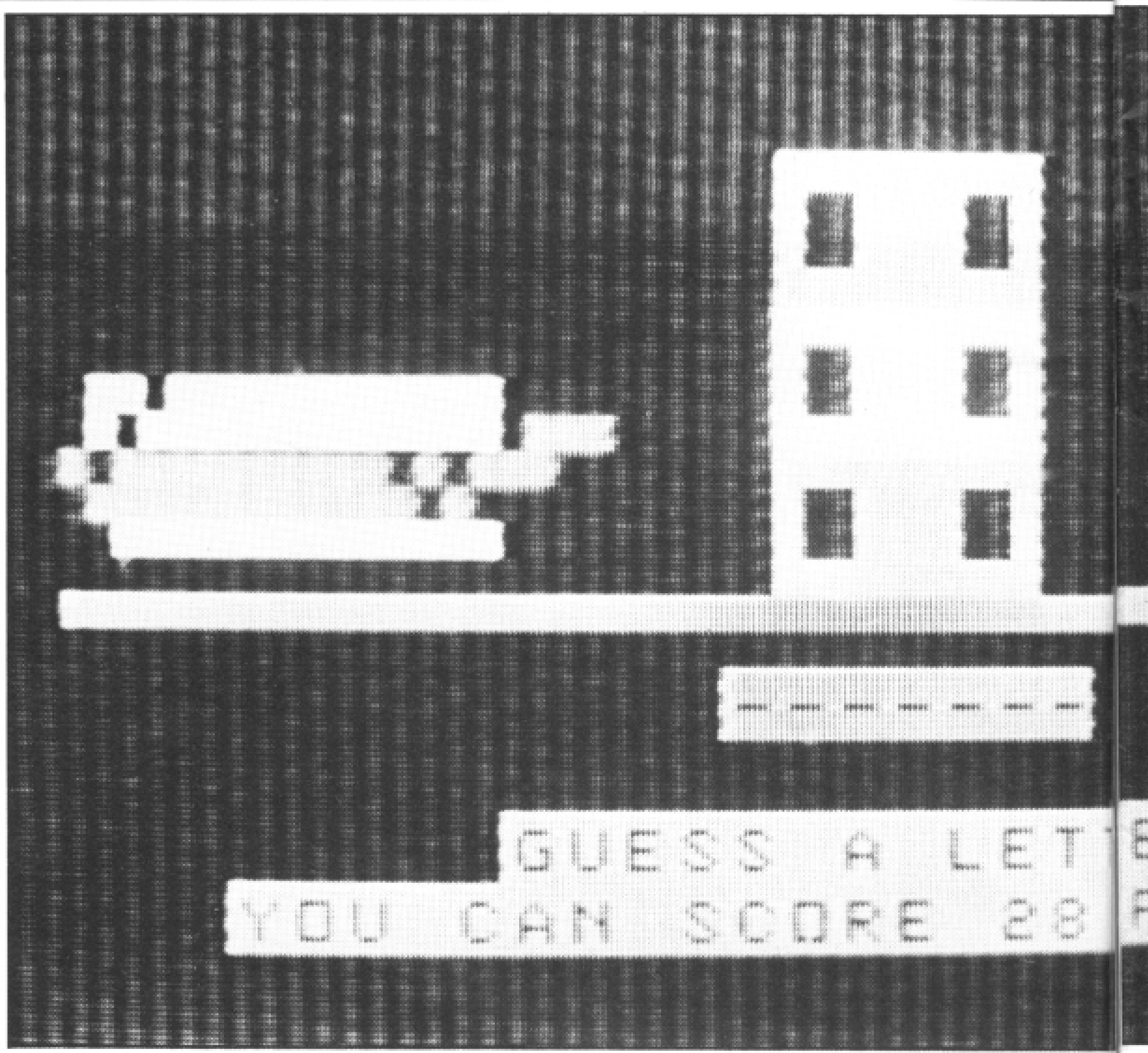
Of course, there is much to learn about computers and the way they work apart from using them to reinforce school work. Circus Adventure (Dragon Data £7.95) sets out with the aim of doing just that job for primary school age, incorporating a number of user inputs "to encourage familiarisation with the keyboard" and presenting the child "with a series of choices to be made". The child who is lost somewhere in the circus has to try to find the popcorn stand. Dragon Data adds that "the average playing time is between 10 to 15 minutes", but I couldn't find anyone to persevere that long with such a boring game.

An adventure game, of necessity, should allow the player to reason out some of the moves or at least get the "you're getting warmer" vibes. No such luck with Circus Adventure. Educationally it is a disaster for it expects children to opt for left and right, north or south without giving any indication of what such concepts mean. The choices it provides are without consequence and meaning. It asks do you want to go UP or DOWN (presumably underground), and treats UP as a mistake for it only wants the first letter. Now every Dragon owner knows that in this case you would use a restricted INKEY $ or use a LEFT $ routine after input but such subtleties are beyond Circus Adventure.

There is no logic in the game plan either. Starting at the entrance and going east leads you to the Tiger's cage, UP (suspended?) is the ticket booth; go right, down and north and you are back at the entrance. The only place this is true is at the north pole. Needless to say this program for young children is entirely in text capitals.

This game bears no comparison to Quest, another Dragon Data adventure. Quest has no pretentions to be educational yet the consequences of moving N, E, S or W are shown on the map. It involves trading and bargaining, the tactics of building up suitable forces and equipment and strategic planning. It has intermediate rewards and is a much better way of introducing adventure games and "computer familiarity". So my advice is to leave Circus Adventure to the monkeys.

Hide and Seek from the same stable is an excellent machine code program written by Applied Systems Knowledge. It uses the full potential of the Dragon's graphics capabilities. It consists of a stimulating series of hide and seek type games



Shards Software's Fun to Learn – runs to 18K and offers five games

of the Kim's Game variety. It firstly familiarises children with the objects to be hidden, encourages matching skills and short term memory, and at the end the association of words with the pictures of the objects they represent. Some important pre-reading skills are incorporated in this superb program which is completely in high resolution colour and uses lower case letters throughout.

Value

There are two programs in the package taking over five minutes to load, 36 very good clear detailed pictures are drawn and a small dictionary is supplied to look up spellings for the final stage. A very nice touch is the use of on-screen symbols to prompt the need for the space bar or re-entry of a word. You can also return to the menu at any time during any six games. Super value for money at £10.95.

Microdeal's Teletutor at £25 is the most expensive of this batch of software. It is packaged in a large ring file with two tapes and pages of detailed documentation.

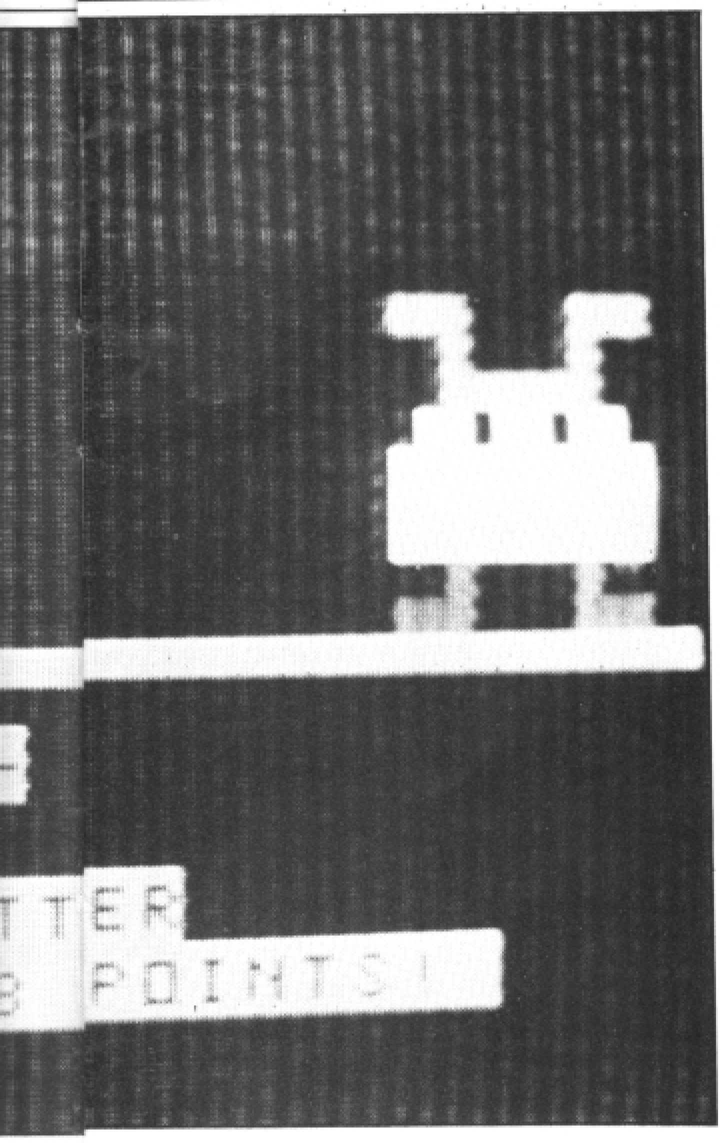
The first program has similar characteristics to Talking Speller (reviewed earlier) except that the time for words to be put onto tape can be varied from between three and 30 seconds per word. Hence sentences showing the word in a sentence could be spoken. You can also get a print-out of results. In other respects this version is not as good as Eurosoft's. You get only one chance at each input (although you can alter words when list has been completed); all answers are accepted and incorrect ones are listed

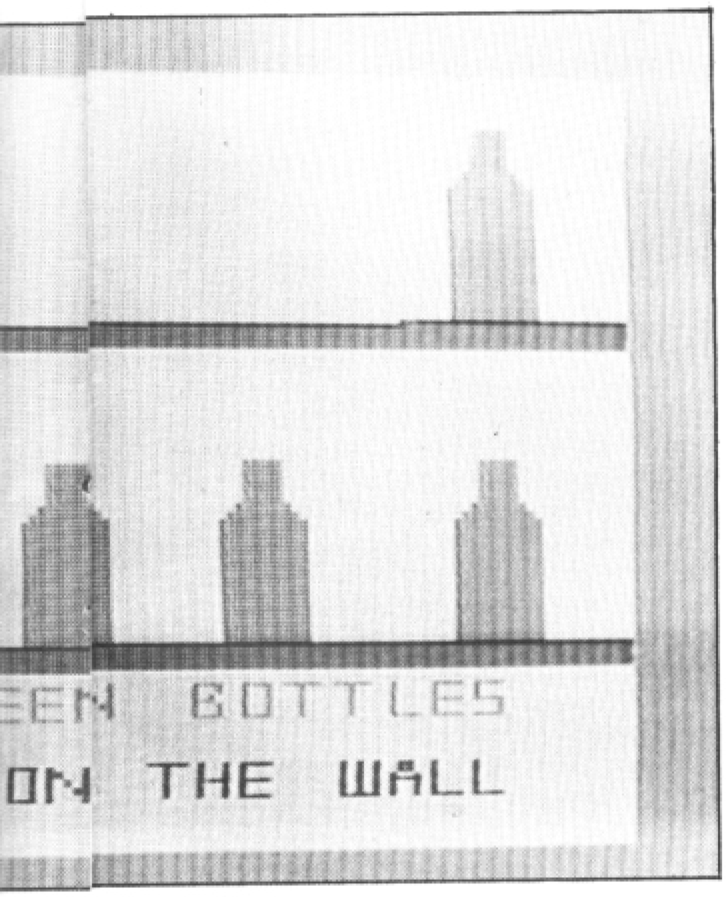


Infant Pack from Shards Software

against the true versions at the end of the test.

Word Drill consists of a series of multiple choice questions in which students are asked to find a word to match definitions. The words and definitions may be added and a data file created or taken from a demo file provided with the package. You could of course use this to test yourself on chemical formulae or make up a geo-





graphical quiz — much in the same way that writing questions on one side of a card and the answer on the other is used as an aid for exam learning. All the words and their definitions can appear on the text screen on command.

Maths Drill allows the digits in answers to computations to be entered right to left as in pencil and paper exercises. In long multiplication questions, partial answers can be entered to build up the sum — but when wrong no attempt is made to analyse why a wrong answer has been reached. In fact when the correct product is printed the figures do not appear in the right columns. A ludicrous reward is offered. It is a glimpse at a hi-res face consisting of a circle line and two dots, the sort of thing you will have first drawn on the hi-res screen when learning.

The fourth program is Estimate which asks a difficult sum and then tells you how far out your guess was.

This set of programs was put together by Tom Mix, the Donkey King man and goes to show just how far behind educational software is compared to games. None of the ingenuity which goes to make up an arcade game (a variety of screens, come-again motivation, fast action and simple aims) is present in this package. It is just dull and stale drill and practice. In fact it is the opposite to what people in education and computing hope for from professional programmers. At this price it is exorbitant.

Micro Debug Consultancy's Tables Tutor at £4.95 follows the same pattern of random number generation that we have seen before. It is child-proofed in as much that the break key has been disabled but your answers can be letters or spurious symbols and these are accepted. In all this is an unenterprising program, children would soon find it boring and it wasn't quite quick enough at displaying question after question to be a real test of tables and skill.

Shards Software has four educational programs ranging from £3.95 to £6.95. Infant Pack purports to teach simple counting and letter recognition skills. In the first program 10 dull looking, unevenly spaced green bottles appear on the screen and are removed one at a time to the accompaniment of that wretched tune. You are then asked 10 times how many bottles remain on the screen (written in capitals), the number being from 1 to 9. The reward for reaching 10 correct answers is much like a nightmare, the screen filling with lines which seem to fill into a murky brown. The reward is best avoided.

On the block

The second program, Alphabet is by far the most appalling educational program I have ever come across. It is supposed to teach letter recognition but draws unrecognisable blocked shapes on the screen purporting to be words. The letters drawn on the screen do not join up at the bottom, the middle prong on capital E is three times as thick as any other, and peculiar square shapes appear in the middle of blocked letters. The words themselves have very uneven spaces between their constituent letters and the pictures drawn on the screen to exemplify the words are little short of pathetic. To cap it all the test at the end doesn't deal with letter recognition at all but asks questions on alphabetical order. It would do the credibility of Shards as a publisher of serious educational software much good if it withdrew this tape from the market immediately.

Junior Pack is a better proposition containing probably the best tables-tester of those on review. You play against the clock and the questions get harder or easier according to your abilities. The accompanying program is more dubious in worth. It is supposed to encourage word recognition skills but the sentences have a random element, eg "D*ve is a strange man. He is a cook" and leave a lot to be desired.

Fun to Learn runs to 18K and is aimed at teaching and reinforcing simple skills in an attractive and entertaining way. Menu driven, the program consists of five games dealing with initial counting, anagrams, simple adding on, a hang-man type game and a series of exercises which encourage word-attack skills through codes. This forms a useful package and should do well at £6.95.

Compendium

Live and Learn is a compendium of programs suitable for juniors and early secondary age children. The first program, Graphics takes you step by step through Dragon high resolution graphics giving examples of each command and using them to create a picture on the screen. You can advance and recap at your own pace. It is disappointing that this is "view only" instruction, no interaction is allowed. It would have been possible to have let the student choose such things as the position of the drawing, the colour of the screen and the size (within limits). This would have been more of a learning experience.

Zoo is a good animal knowledge program. It contains information on 20 animals and for each one a hi-res map can be called on to show its world-wide distribution. There is a "guess the animal" quiz option — you have to be able to spell the names correctly for the computer to recognise your answer.

With Britain the well-drawn map fooled me into high expectations but all it had to offer in the end was a list of the characteristics of countless towns in England, Wales and Scotland. It offers a good example of the abuse of the sound function. Imagine 30 or so pages of text and each letter accompanied by a piercing bleep. I felt I had been dragged around Britain by the ears.

Live and Learn ends with Survivor which is an interesting and safer method of testing your chances of desert survival than the real thing — and Music which consists of a simple tutorial followed by your chance to do a Vangelis on the Dragon keyboard.

So how does your excuse of the lack of educational software hold up now? I have looked at some awful programs, some mediocre and a few good ones. Do any of them suit your needs, and how do you choose? Any serious educational publisher should produce some documentation to go with the software. Schools often select packages by sending first for this literature. They can then make some judgement on its value and decide if it suits their needs. Perhaps parents with home computers should do the same thing. ■

Putting some snap into learning maths

***Ged Mead** shows you how you can encourage your children to learn their times tables – and have some fun into the bargain*

HOME COMPUTERS can be great fun for playing games, but many people buy a micro to use it, among other things, as a teaching aid. It is quite a convincing reason in those early days when you are trying to convince your wife, friends (or even yourself!) that you have made a worthwhile investment.

Your micro does have the potential to make learning fun for children, but unfortunately most of the listings in magazines and books are for games. Consequently it can be quite difficult to take advantage of this useful feature of your computer without guidance. This article should help the less experienced programmer — it shows how to construct a simple maths game in which two children compete to be the first to spot the correct answer to sums displayed on the screen.

Starting . . .

Type in the lines of the program listed with this article in the order they are discussed here. To help you, the various sections of lines are italicised to show when you should add them to your listing.

Start with *lines 40-90* which identify the names of the players and give them their instructions. Line 90 uses the INKEY$ function to keep the instructions on screen until the players are ready to move on. As soon as C$ has a value (that is, when any key is pressed) the program can move on to the next line.

Line 110 generates the sums to be answered. The variable A will have a value between 3 and 12. "A=RND(10)+2" can be thought of as "pick a number from 1 to 10, then add 2 to it", so the range will be from 3 (ie, 1+2) to 12 (10+2). Similarly, B will vary between 4 and 12. C is the result of multiplying A and B.

To produce an "answer" to tempt our young players with, we use the useful statement in *line 170,* which might need a little explanation. Let us assume, for example's sake, that A=9 and B=6, in which case C will of course be 54. Now if you think of "D=RND(3)+(C−2)" as meaning "take a number between 1 and 3 (1, 2 or 3); add the value of C to it (making 55, 56 or 57 in our example); then take away 2 from this total" you will see that the three possible numbers produced are either 53, 54 or 55. This will work for any value of C that our program generates, producing a number which is equal to, one less or one more than C itself.

Line 150 now prints the question, *160* builds in a variable pause and *190* prints the "answer" offered, together with a sound prompt to draw attention to it.

Once the "answer" is on the screen we expect the players to react by pressing a key if the correct answer is being shown. To keep competitiveness and interest at a maximum we will construct our program so that if both children press their key then both presses will be recorded and suitable messages displayed. The quicker child will still win, but the slower one will also earn praise if the right key presses are made.

To achieve this, we have to adapt the INKEY$ function to suit our purposes. The standard INKEY$ statement (eg B$=INKEY$) alone will only record the value of the last key pressed — and would actually identify the slowest player as the winner! So we will employ a string array in which we can store the various key presses made. When a key is depressed, its value (which should be the letter "A" or "L" if the players have followed their instructions) will be stored in the array B$(). By this method, the first key press can be stored in B$(0) and the second (if any) in B$(1), so we can easily evaluate who was the fastest.

Enter *lines 200-240.* Line 200 sets up the INKEY$ function, and line 210 a FOR . . . NEXT loop which effectively scans the keyboard up to 100 times. The first statement in line 220 introduces the string array and means "when a key is pressed, store its value in the next subscript of the array B$". Until a key is pressed J will equal 0, so the first key press will be stored in B$(0). In order to keep the program circling through our FOR . . . NEXT loop if no key has been pressed, we add the second statement in 220 which reads as "if no letter is yet stored in the current subscript of B$() then go directly to line 240", from where the loop will restart.

If, on the other hand, a key has been pressed since the last time the INKEY$ checked the keyboard then B$(J) will have a value (the letter pressed) and will NOT equal "". It will therefore ignore the IF . . . THEN statement and drop through to line 230.

If line 230 is reached it must be because BS(0) has a value stored in it. As this subscript is now "occupied" we add 1 to the value of J so that the next key press made can be put into BS(1). The second statement in this line simply ensures that once two keys have been pressed the program will jump out of the FOR . . . NEXT loop, moving the game into the next stage.

Line 250 will only be reached if J=0 (no keys pressed during the whole run of the FOR . . . NEXT loop) or J=1 (only one key pressed). If J does equal 1 the program jumps to line 300. If J=0 then *line 260* will assess whether keys should have been pressed or not. If the answer offered on screen was not the right one the program moves to *line 290,* gives a short prompt then goes back to line 120 to start the sequence again with a new value for D. If the right answer was displayed (ie, D=C) then *line 270* points out the missed opportunity to our two budding mathematicians. *Line 280* pauses, then sends control back to line 100 from where a new series of variables are created.

. . . block

The next block of lines from *300-450* cover the permutations when only one key has been pressed. Lines 310-370 apply if the right answer was displayed and will congratulate the quick-witted player who pressed, then go to the appropriate subroutines in *lines 930-940* which keep score. If neither "A" nor "L" were pressed then this is picked up in line 370 and *line 750* prints a message accordingly. In all cases, the program then goes back to lines 280 and 100 to restart the sequence.

Lines 400-450 point out the error of his ways to a player pressing when he ▶

```
10 REM **MATHSNAP! BY GD MEAD**
30 E=0:F=0
40 CLS:PRINT@ 70,"££££  MATHSNAP!  ££££":PRINT:PRINT
50 INPUT "FIRST PLAYER'S NAME";A$:INPUT"SECOND PLAYER'S NAME";L$
70 CLS:PRINT:PRINT"WHEN YOU SEE THE CORRECT ANSWER":PRINT"TO THE SUM PRINTED ON
SCREEN":PRINT"PRESS YOUR OWN KEY once ONLY":PRINT"AS QUICKLY AS YOU CAN!":PRINT
80 PRINT A$;" - USE THE 'A' KEY":PRINT L$;" - USE THE 'L' KEY":PRINT:PRINT"THE F
IRST PLAYER TO SCORE 25":PRINT"POINTS WILL WIN THE ROUND"
90 PRINT @ 448, "PRESS ANY KEY TO START PLAY":C$=INKEY$:IF C$="" THEN 90
100 M=0
110 A=RND(10)+2:B=RND(9)+3:C=A*B
120 FOR J=0 TO 3:B$(J)="":NEXT J:J=0
130 CLS:PRINT@12,"scores":PRINT@32,A$:PRINT@48,L$:GOSUB 980:PRINT:PRINTSTRING$(3
2,134)
140 IF E>=25 OR F>=25 THEN 840
150 PRINT@198,A;" X ";B;" = "
160 FOR K=1TO RND(500)+300:NEXT K
170 D=RND(3)+(C-2)
180 IF D<>C THEN M=M+1:IF M=4 THEN D=C
190 PRINT@263,D;"  ....??":SOUND 210,2:PRINT
200 B$=INKEY$
210 FOR K=1TO100:B$(J)=INKEY$
220 IF B$(J)=""THEN 240
230 J=J+1:IF J=2 THEN 460
240 NEXT K
250 IF J=1 THEN 300
260 IF D<>C THEN 290
270 PRINT"MISSED YOUR CHANCE!":SOUND 1,4:PRINTD;"is EQUAL TO";A;" X";B
280 FOR K=1 TO 1200:NEXT K:GOTO 100
290 PRINT@448,"READY?...":SOUND 160,4:GOTO120
300 IF D<>C THEN 400
310 PRINT"TIME UP!":SOUND 1,4:PRINT"KEY CORRECTLY PRESSED BY:- "
320 IF B$(0)="A" THEN PRINT A$:GOSUB930:GOTO280
340 IF B$(0)="L" THEN PRINT L$:GOSUB940:GOTO280
370 GOSUB 750:GOTO 280
400 PRINT"NO! YOU SHOULD NOT HAVE PRESSED"
410 IF B$(0)="A" THEN PRINT A$:GOSUB950:GOTO450
420 IF B$(0)="L" THEN PRINT L$:GOSUB 960:GOTO450
430 PRINT"BUT IN ANY CASE......."
440 SOUND 20,3:PRINT"WRONG KEY PRESSED"
450 FOR K=1TO1000:NEXTK:GOTO290
460 IF B$(0)<>B$(1) THEN 520
470 SOUND 20,5:PRINT"SAME KEY PRESSED TWICE"
480 IF B$(0)="A" THEN PRINT A$:GOSUB940:GOTO280
```

◀ shouldn't; *lines 950 and 960* reduce the score. Again, if the key was not "A" or "L" this is dealt with in line 430.

Lines 460-690 come into play when two keys have been pressed (check back to line 230 if you are unsure about this). First of all we have to discourage the smart Alec who might try and get extra points by pressing his own key twice so as to exclude the other player. This is done by *lines 460 to 490* which rap the scoundrel's knuckles and then increase the other player's score. *Line 500* only reacts if the twice pressed key was neither "A" nor "L".

Once this has been checked out, *lines 520-610* sort through the contents of B$() to decide who was first (the player whose letter is found in B$(0)), second (B$(1)), and whether any other keys apart from allowed ones were pressed. Although lines 530-610 may seem complex at first, if you have stayed with me so far and understood how the earlier lines 300-450 did their job, you should have little difficulty seeing how this section carries out its tasks too. Lines 530-560 analyse the first press and 580-610 process the second.

In *lines 630-700* action is taken if both players pressed when they shouldn't have. In this case it doesn't really matter who pressed first as both players are penalised equally, but as we have the information we may as well display it on screen! Lines 660 and 700 detect if an invalid key was pressed.

Loose ends

Now to tidy up a few loose ends. *Line 30* resets the scores to zero at start of play. *Line 120* ensures that the array B$ is empty before each new display is made (if we didn't do this, late presses made during one display might be read as a very fast press on the next). *Line 140* checks the scores and jumps to the "winner" sequence if either player has reached 25 points.

The sections we have covered so far are the backbone of the program. The remaining lines add some sound and action. *Line 130* is a simple screen display of the players' names and scores. *Lines 760-790* POKE a line of asterisks over Player 1's name if he scores (or if the other player is penalised) and *line 980* updates the score on screen. *Lines 800-830* do the same for the other player. In *lines 840-890* the winner is announced with a small fanfare and flashing lights.

Line 180 prevents the game from getting hung up on one sum by counting how many consecutive wrong answers are generated in line 170. It then ensures that by the fourth display the right answer will be offered. *Line 100* resets this counter to zero.

Depending on the children involved, the pleasure of being the winner may be reward enough — but in my experience this won't last long and more enticement may be necessary to keep them at it. One good way of achieving this is to let the winner play a round or two of an arcade-type game before the program reverts to the question and answer routines. You can use any game program written in Basic for this (*try Tracker, elsewhere in this issue*), as explained below.

Enter *lines 900-920*, filling in the name of the arcade game. The demand for the winner's name is really only a bit of showmanship for the winner and could easily be by-passed.

Now carry out the following steps:

```
490 IF B$(0)="L" THEN PRINT L$:GOSUB 930:GOTO280
500 GOTO430
520 IF D<>C THEN 630
530 PRINT"FIRST KEY PRESSED WAS.."
540 IF B$(0)="A" THEN PRINT A$:E=E+2:GOSUB 760:GOTO590
550 IF B$(0)="L" THEN PRINT L$:F=F+2:GOSUB 800:GOTO580
560 GOSUB 750
580 IF B$(1)="A" THEN  PRINT @384,A$:E=E+1:GOSUB 760:GOTO610
590 IF B$(1)="L" THEN PRINT@384, L$:F=F+1:GOSUB 800:GOTO610
600 PRINT@384,"SECOND KEY...":GOTO 370
610 PRINT@416,"WAS THE SECOND TO PRESS":GOTO280
630 PRINT"YOU ARE BOTH WRONG"
640 IF B$(0)="A" THEN PRINT A$:GOSUB 950:GOTO670
650 IF B$(0)="L" THEN PRINT L$:GOSUB 960:GOTO670
660 GOSUB 750
670 PRINT@384,"AND SECOND PRESS WAS.."
680 IF B$(1)="A" THEN PRINT@416, A$:GOSUB 950:GOTO290
690 IF B$(1)="L" THEN PRINT@416, L$:GOSUB 960:GOTO290
700 GOSUB 750:GOTO280
750 SOUND 15,4:PRINT"WRONG KEY PRESSED":RETURN
760 FOR Z=0TO7
770 POKE 1024+Z,42:SOUND 200,2
780 NEXT Z
790 GOSUB 980:RETURN
800 FOR Z=0TO7
810 POKE 1055-Z,42:SOUND 200,2
820 NEXT Z
830 GOSUB 980:RETURN
840 CLS 3:PRINT@96," "
850 FOR Z=1TO6:SOUND 140,3:PRINT@96,"   ££££  A WINNER!!   ££££":NEXT Z
860 PRINT:PRINT:PRINT"AND THE WINNER IS ....":PRINT
870 PLAY "T18V25O4GL2CO3CDDFCCCFDECCO4GL2C"
880 IF E>F THEN PRINT A$:W$=A$ ELSE PRINT L$:W$=L$
890 FOR K=1TO700:NEXT K
900 CLS:PRINT"THE WINNER OF THIS ROUND":PRINT"HAS EARNED A GAME":PRINT"OF ******
******** !":PRINT"TYPE IN THE WINNER'S NAME TO":PRINT"START THE GAME - OR TYPE I
N THE":PRINT"WORD 'MATHS' FOR ANOTHER GAME":PRINT"OF mathsnap":PRINT
910 INPUT "NAME.....";X$
920 IF X$=W$ THEN 1000 ELSE IF X$="MATHS" THEN 30 ELSE  SOUND 10,7:GOTO 900
930 E=E+3:GOSUB760:RETURN
940 F=F+3:GOSUB 800:RETURN
950 E=E-1:GOSUB 800:RETURN
960 F=F-1:GOSUB 760:RETURN
980 PRINT@72,E:PRINT@88,F:RETURN
```

1. CSAVE "Mathsnap" on to a tape.
2. CLOAD your chosen arcade game.
3. RENUM the arcade game, making the first line number 1000 — see page 43 of Dragon handbook.
4. CSAVE the now renumbered arcade game on tape.
5. CLOAD "Mathsnap"
6. Use the Immediate mode (ie, type in these instructions directly without line numbers, then press "ENTER"). The quotation marks are for clarity only — don't type them in. Type "PRINT PEEK(25),PEEK(26)" and then press ENTER.
7. Make a note of the two numbers that appear on screen.
8. Type "PRINT PEEK(28)" and ENTER.
9. If this latest number is greater than 1 then type "POKE 25, PEEK(27) : POKE 26, PEEK(28)−2". Otherwise type "POKE 25,PEEK(27)−1 : POKE 26, 256−PEEK(28)".
10. CLOAD the renumbered arcade game.
11. In the immediate mode, type "POKE 25," and then type in the first number from instruction 6. ENTER this. Now type "POKE 26," and the second number from instruction 6. ENTER this.

If you have followed the above instructions exactly you will have merged the two programs into one.

All that remains is for us to add the lines that will ensure that once our winner has had the allotted period on the arcade game the program will return to the maths game. This can be done in a number of ways and I offer you two.

Back to maths

Firstly, find the line in the arcade game (if any) that gives the player the choice of playing again. Alter this line to read INPUT "ANOTHER GAME (Y/N);Z$:IF Z$="Y" THEN RUN ELSE END".

Alternatively, use the TIMER function and insert an additional line in the maths program:

```
915 TIMER=0
```

Find a line in your arcade game which is often actioned as the program runs (eg, one that updates the score) and insert these two statements in it T1=TIMER : IF T1>7000 THEN 50000. Finally, add new line 50000:

```
50000 CLS:PRINT "TIME UP";W$ : FOR
      K=1 TO 1000 : NEXT : GOTO 30
```

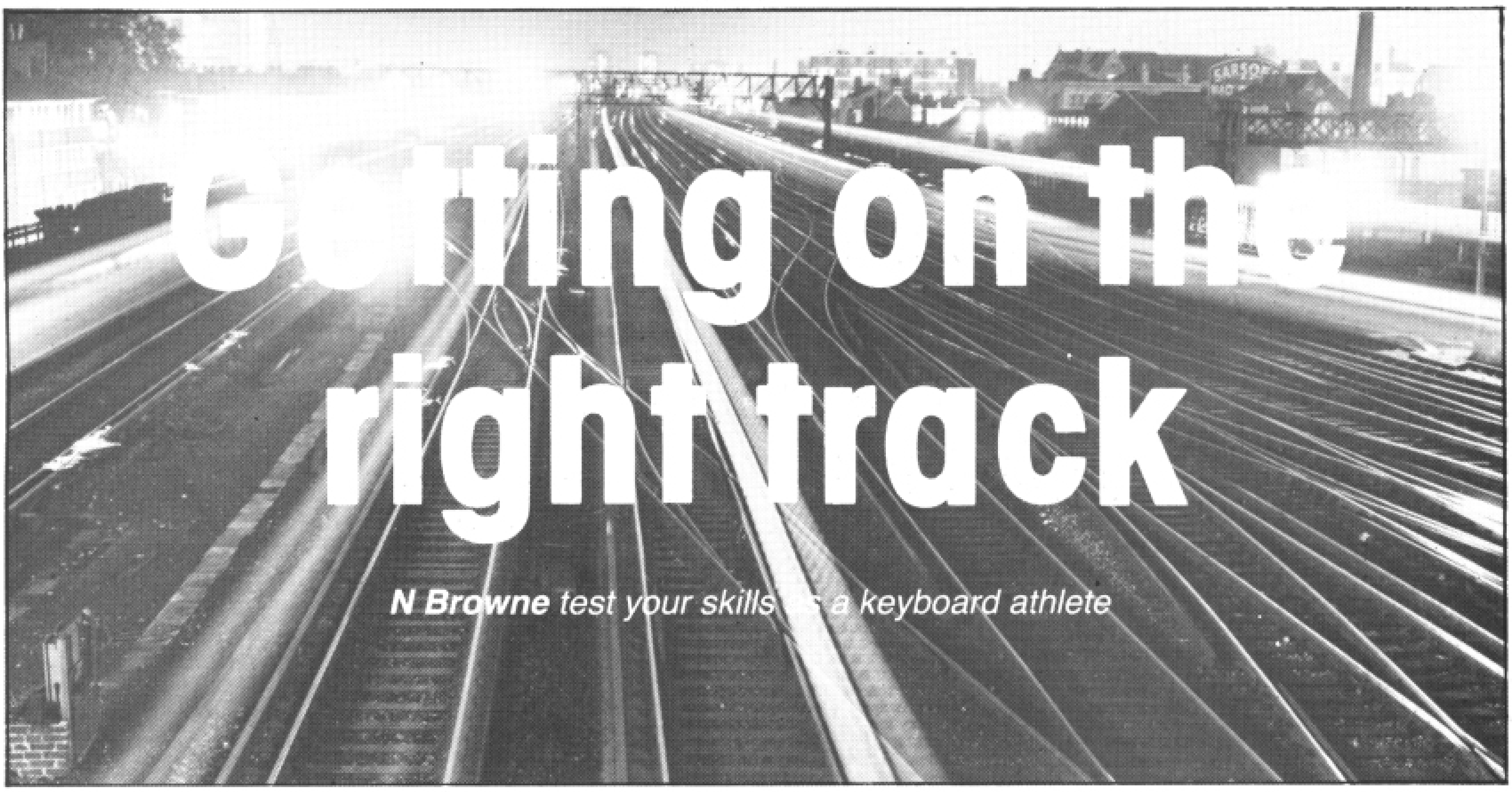
I'll close by offering some variations. Add the following lines:

```
162 S1=RND(2) : ON S1 GOTO 164,170
164 S2=RND(3) : IF S2=1 THEN
    D=C−B : GOTO 180 ELSE IF S2=2
    THEN D=C+B : GOTO 180 ELSE
    D=C : GOTO 190
```

which will randomly produce a different series of possible "answers". Increase display time (reaction time) by increasing the number in line 210. To have a new sum generated each display, amend the last figure in line 290 from "120" to "100". Alter display time of screen messages by changing the numbers in lines 280 and 450.

You have there the makings of a first-class competitive maths game — it's over to you now to enhance it with sound and colour, animation and variety. Remember the two key points for a successful children's program — keep it lively and always reward a good effort!

If you want to contact me, write to Thatchover Cottage, School Lane, Middleton Stoney, Oxon OX6 8SW. ■



Getting on the right track

N Browne test your skills as a keyboard athlete

```
10 '********** TRACKER **********
20 '*                           *
30 '* COPYRIGHT N.A.BROWNE 1983 *
40 '*                           *
50 '*****************************
60 '
70 POKE&HFFD7,0
80 GOSUB1270
90 DIMX(6),Y(6),XI(6),YI(6),B$(15),D(10)
100 POKE359,60:P=0
110 CLS:PRINT:PRINTTAB(13)"tracker"
120 PRINT,," THE OBJECTIVE IS TO KEEP THE     TRAINS (BLUE DOTS) MOVING ON     THE
 YELLOW TRACKS."
130 PRINT" OPERATE THE JUNCTIONS BY USE     OF THE ARROW KEYS. YOU STOP      MOVEM
ENT IN THE DIRECTION OF     THE ARROW KEY. FOR EXAMPLE '^'"
140 PRINT" STOPS MOVEMENT VERTICALLY."
150 PRINT,," ENTER NUMBER OF TRAINS (2-6)"
160 PRINT" OR 'D' FOR DEMONSTRATION MODE    OR 'E' TO END."
170 POKE359,57
180 A$=INKEY$:IFA$<>""THEN210
190 FORX=1TO200:NEXT:PRINT@45,"TRACKER":FORX=1TO200:NEXT:PRINT@45,"tracker"
200 GOTO180
210 D=VAL(A$):IFD>6THEN180
220 IFD>1THENC=0:GOTO260
230 IF A$="D"THEN D=5:C=1:GOTO260
240 IF A$<>"E"THEN180
250 CLS:PRINT:PRINTTAB(7)"PROGRAM TERMINATED":PRINT:POKE&HFFD6,0:END
260 PMODE1,1:PCLS1:SCREEN1,0:COLOR2,1
270 LINE(10,8)-(246,184),PSET,B:LINE(10,8)-(122,120),PSET,B:LINE(122,60)-(246,18
4),PSET,B
280 LINE(58,10)-(58,120),PSET:LINE(186,60)-(186,184),PSET:LINE(0,56)-(10,56),PSE
T
290 COLOR3,1
300 GOSUB870
310 FORZ=1TO6:XI(Z)=4:YI(Z)=0:X(Z)=2:Y(Z)=56:NEXT
320 GOSUB570
330 TIMER=0:Q=0:R=0:T=0
340 FORV=1TOD
350 A$=INKEY$:IFA$="^"THENGOSUB600
360 IFA$=CHR$(10)THENGOSUB600
370 IFA$=CHR$(8)THENGOSUB600
380 IFA$=CHR$(9)THENGOSUB600
390 IFA$="E"THENRUN
400 T=T+1:IFT=4THENGOSUB1200
410 FORZ=1TOV
420 X(Z)=X(Z)+XI(Z):Y(Z)=Y(Z)+YI(Z)
430 'IFPPOINT(X(Z),Y(Z))=3THEN760
440 IFPPOINT(X(Z),Y(Z))=1THENX(Z)=X(Z)-XI(Z):Y(Z)=Y(Z)-YI(Z):IFXI(Z)=0THEN490ELS
E530
450 IFPPOINT(X(Z)-XI(Z),Y(Z)-YI(Z))=1THENN=1ELSEN=2
460 PSET(X(Z)-XI(Z),Y(Z)-YI(Z),N):PSET(X(Z),Y(Z),3)
470 NEXT:NEXT
480 GOTO340
490 IFPPOINT(X(Z)+3,Y(Z))=2ORPPOINT(X(Z)+3,Y(Z))=3THENXI(Z)=4:YI(Z)=0:GOTO420
500 IFPPOINT(X(Z)-3,Y(Z))=2ORPPOINT(X(Z)-3,Y(Z))=3THENXI(Z)=-4:YI(Z)=0:GOTO420
510 IFC=1THENGOSUB570:GOTO420
520 GOTO760
530 IFPPOINT(X(Z),Y(Z)+2)=2ORPPOINT(X(Z),Y(Z)+2)=3THENYI(Z)=4:XI(Z)=0:GOTO420
540 IFPPOINT(X(Z),Y(Z)-2)=2ORPPOINT(X(Z),Y(Z)-2)=3THENYI(Z)=-4:XI(Z)=0:GOTO420
550 IFC=1THENGOSUB570:GOTO420
560 GOTO760
570 M=RND(4)+7:IFM=11THENM=94
580 IF M=E THEN570ELSEE=M
590 A$=CHR$(M)
600 SOUND200,1:IFA$="^"THENN=1ELSEN=2
610 COLOR3,1:LINE(180,158)-(174,110),PRESET:LINE-(146,110),PRESET
620 LINE(146,110)-(156,130),PSET:LINE-(160,110),PSET:LINE-(180,158),PSET
```

THIS NEW game for the Dragon is written entirely in Basic but it's fast enough to outpace the best qwerty keyboard athlete. It is compatible with both colour, and black and white televisions.

Tracker is a game loosely based on a railwayman's tracker board and evolved from a simulation-type program. The tracker board maintains the positions, speed of trains and the status of the junctions along each railway line. Here trains are represented by blue dots and the railway lines as a grid system of yellow tracks. Your objective is to keep all the trains on the tracks for as long as possible.

A train moving towards a junction with a break in the track ahead implies the points are set against it and only the skilful use of the arrow keys will prevent the pending disaster. Points are reset by pressing an arrow key in the direction you wish the train not to travel. For example, there's a train moving down the screen with the points set against it, you press any arrow key other than up to stop a derailment.

When an arrow key is pressed the simple graphic signalman will pull his lever and the points reset. But just to make life difficult, each time one set of points are changed, all the others change in a similar manner. There are two concessions, though, trains may overtake and pass from opposite directions. I didn't want to make the game totally impossible.

A record of time, in approximate seconds, is kept and constantly updated on the screen as well as the current best time (labelled BT).

The program begins with a colourful Tracker logo display followed by a brief text of instructions. The user is requested to select between two and six trains or opt for the demonstration mode. I suggest you select two trains at first until you are familiar with the method of resetting the points. In the computer demonstration mode the computer plays using five trains and never loses track of any of them. ▶

```
630 PSET(10,118,N):PSET(10,116,N):PSET(58,118,N):PSET(58,116,N):PSET(122,58,N):P
SET(122,56,N):PSET(122,118,N):PSET(122,116,N)
640 PSET(246,58,N):PSET(246,56,N):PSET(122,182,N):PSET(122,180,N):PSET(186,182,N
):PSET(186,180,N)
650 IFA$=CHR$(10)THENN=1ELSEN=2
660 PSET(58,10,N):PSET(58,12,N):PSET(122,10,N):PSET(122,12,N):PSET(122,62,N):PSE
T(122,64,N)
670 PSET(122,122,N):PSET(122,124,N):PSET(10,122,N):PSET(10,124,N):PSET(186,62,N)
:PSET(186,64,N):PSET(246,62,N):PSET(246,64,N)
680 IFA$=CHR$(8)THENN=1ELSEN=2
690 PSET(56,8,N):PSET(54,8,N):PSET(120,8,N):PSET(118,8,N):PSET(56,120,N):PSET(54
,120,N):PSET(184,60,N):PSET(182,60,N)
700 PSET(244,60,N):PSET(242,60,N):PSET(120,120,N):PSET(118,120,N):PSET(120,184,N
):PSET(118,184,N):PSET(184,184,N):PSET(182,184,N)
710 IFA$=CHR$(9)THENN=1ELSEN=2
720 PSET(60,8,N):PSET(62,8,N):PSET(124,8,N):PSET(126,8,N):PSET(124,60,N):PSET(12
6,60,N):PSET(188,60,N):PSET(190,60,N)
730 PSET(60,120,N):PSET(62,120,N):PSET(12,120,N):PSET(14,120,N):PSET(124,184,N):
PSET(126,184,N):PSET(188,184,N):PSET(190,184,N)
740 LINE(146,110)-(156,130),PRESET:LINE-(160,110),PRESET:LINE-(180,158),PRESET:L
INE(146,110)-(174,110),PSET:LINE-(180,158),PSET
750 RETURN
760 SOUND1,10:FORM=0TO16STEP4:SOUND32-M,1:CIRCLE(X(Z),Y(Z)),M,4:NEXT
770 IFZ<V THENZ=V:NEXTZ
780 IFV<D THENV=D:NEXTV
790 IFC=1THEN850
800 P=(R*100)+(Q*10)+P
810 IFP>S THENS=P
820 R=INT(S/100):Q=INT((S-(R*100))/10):P=S-(R*100)-(Q*10)
830 PUT(78,52)-(112,42),D
840 DRAW"BM78,52;C4"+B$(R)+B$(Q)+B$(P)
850 FORM=1TO5000:NEXT:IFC=1THENRUN
860 SCREEN0,0:GOTO180
870 B$(10)="BM+2,+0;U8L4R8"
880 B$(2)="BM+6,+8;R4L2U8L2R4"
890 B$(3)="BM+6,+8;U8F4E4D8"
900 B$(11)="BM+6,+0;NR6U4NR4U3R6"
910 B$(5)="BM+6,+8;U2BM+0,-5;U1"
920 DRAW"BM144,36;"+B$(10)+B$(2)+B$(3)+B$(11)+B$(5)
930 B$(0)="BM+6,+0;L6U8R6D8BM+4,0;"
940 B$(1)="BM+2,+0;U8BM+6,+8;"
950 B$(2)="R6L6U4R6U4L6BM+10,+8;"
960 B$(3)="R6U4L6R6U4L6BM+10,+8;"
970 B$(4)="BM+6,+0;U8G6R8BM+4,+2;"
980 B$(5)="R6U4L6U4R6BM+4,+8;"
990 B$(6)="U8D8R6U4L6BM+10,+4;"
1000 B$(7)="BM+6,+0;U8L6BM+10,+8;"
1010 B$(8)="R6U8L6D8U4R6BM+4,+4;"
1020 B$(9)="BM+6,+0;U8L6D4R6BM+4,+4;"
1030 B$(12)="BM+4,+8;U8R6D4L6R2F4"
1040 B$(13)="BM+4,+0;U4R6L6U4R6D8"
1050 B$(14)="BM+4,+0;R6L6U8R6"
1060 B$(15)="BM+4,+8;U8D6E6G4F4"
1070 DRAW"BM32,154;"+B$(10)+B$(12)+B$(13)+B$(14)+B$(15)
1080 DRAW"BM76,154;"+B$(11)+B$(12)
1090 PUT(78,52)-(112,42),D
1100 DRAW"BM78,52;C4"+B$(R)+B$(Q)+B$(P)
1110 COLOR3,1:LINE(30,158)-(100,158),PSET
1120 CIRCLE(146,90),10,3:LINE(146,100)-(146,130),PSET:LINE-(134,160),PSET:LINE(1
46,130)-(158,160),PSET
1130 CIRCLE(134,158),4,3:CIRCLE(158,158),4,3:LINE(146,110)-(174,110),PSET:PSET(1
44,88,4):PSET(148,88,4):PSET(146,92,4):LINE(146,110)-(128,128),PSET
1140 LINE(174,110)-(180,158),PSET:CIRCLE(178,158),4,3
1150 LINE(142,74)-(150,80),PSET,B:LINE(140,80)-(152,80),PSET
1160 DRAW"BM78,36;U8R4F2G2L4R4F2G2L4R4BM+14,+0;"+B$(10)
1170 PSET(88,36,4):PSET(104,36,4)
1180 DRAW"BM16,56;R8D12H4F4E4"
1190 RETURN
1200 P=INT(TIMER/60):T=0
1210 IFP>9THEN Q=INT(P/10):P=P-(Q*10)
1220 IFQ>9THEN R=INT(Q/10):Q=Q-(R*10)
1230 IFR>9THENTIMER=0:Q=0:R=0:GOTO1200
1240 PUT(200,26)-(234,36),D
1250 DRAW"BM200,36;C4"+B$(R)+B$(Q)+B$(P)
1260 RETURN
1270 CLS0:FORP=0TO112STEP16:PRINT@0,STRING$(5,CHR$(131+P));
1280 PRINT@34,CHR$(143+P);:PRINT@66,CHR$(143+P);CHR$(128);CHR$(128);CHR$(143+P);
CHR$(140+P);CHR$(143+P);
1290 PRINT@101,CHR$(143+P);CHR$(131+P);CHR$(143+P);:PRINT@133,CHR$(143+P);CHR$(1
28);CHR$(137+P);CHR$(128);CHR$(131+P);CHR$(131+P);CHR$(131+P);
1300 PRINT@169,CHR$(143+P);CHR$(128);CHR$(143+P);:PRINT@201,CHR$(143+P);CHR$(140
+P);CHR$(143+P);CHR$(128);STRING$(4,CHR$(131+P));
1310 PRINT@237,CHR$(143+P);:PRINT@269,CHR$(143+P);:PRINT@301,STRING$(4,CHR$(140+
P));CHR$(128);CHR$(143+P);CHR$(128);CHR$(134+P);
1320 PRINT@233,CHR$(140+P);CHR$(128);CHR$(140+P);
1330 PRINT@338,CHR$(143+P);CHR$(143+P);:PRINT@370,CHR$(143+P);CHR$(128);CHR$(137
+P);CHR$(128);CHR$(143+P);CHR$(140+P);CHR$(140+P);CHR$(140+P);
1340 PRINT@406,CHR$(143+P);CHR$(140+P);CHR$(140+P);:PRINT@438,CHR$(143+P);CHR$(1
31+P);CHR$(131+P);CHR$(131+P);CHR$(128);CHR$(143+P);CHR$(140+P);CHR$(143+P);
1350 PRINT@475,CHR$(143+P);CHR$(131+P);CHR$(143+P);:PRINT@507,CHR$(143+P);CHR$(1
28);CHR$(137+P);
1360 FORZ=1TO1000:NEXT
1370 NEXT:RETURN
```

◀ To end the demonstration mode press E at any time.

The speed-up POKE is used in line 70, but it is not essential.

Program notes

Lines	
70	Utilisation of the high-speed poke, although its use is not essential to the game.
90	Dimensions the arrays.
100	A poke to slow down the printing to screen speed.
110-240	Displays the text of instructions, selects number of trains or demonstration mode and returns screen printing to normal speed.
250	Is the end of program routine which pokes the computer back to normal speed.
260-290	Sets the high-resolution parameters, the background, foreground colours and draws the railway lines.
310	Initialises the train movement variables.
330	Zeroes the timer variables.
350-380	Checks for arrow key use.
410-560	Controls train movements and checks for derailments.
600-750	Resets the junction points and makes the signalman pull the lever.
760-860	The derailment routine and checks for and displays a best time.
870-1190	The draw string commands for the remainder of the high-resolution display.
1200-1260	Updates the timer and displays the time. Should the timer exceed 999 it zeroes.
1270-1370	The Tracker logo display routine.

Program variables

Y(Z), X(Z)	Stores the position of each train.
YI(Z), XI(Z)	Stores the direction of movement of each train.
C	Computer demonstration mode flag (C=1 on, C=0 off).
P, Q, R	Timer variables — seconds, 10 seconds and 100 seconds respectively.
S	Best time.
D, D(10)	Array for blanking out the numbers of the timer and best time.
N	Stores the colour for the PSETs used in resetting the points (1 green, 2 yellow).
M	Stores the initial random points setting and the demonstration mode points resetting.
E	Prevents the computer, in demonstration mode, from resetting the previous points setting.
B$(1) – B$(15)	Stores the draw string commands for the numbers and text on the high-resolution screen.

It is not necessary to GET a blank part of the screen for blanking out displays.

Masochists should try the program after activating line 430 by removing the initial quote. This will have the effect of preventing trains from overtaking or passing from opposite directions. When a collision is detected processing is transferred to the derailment routine. It also effects the demonstration mode since there is no check for trains on the same line. The computer will now lose trains.

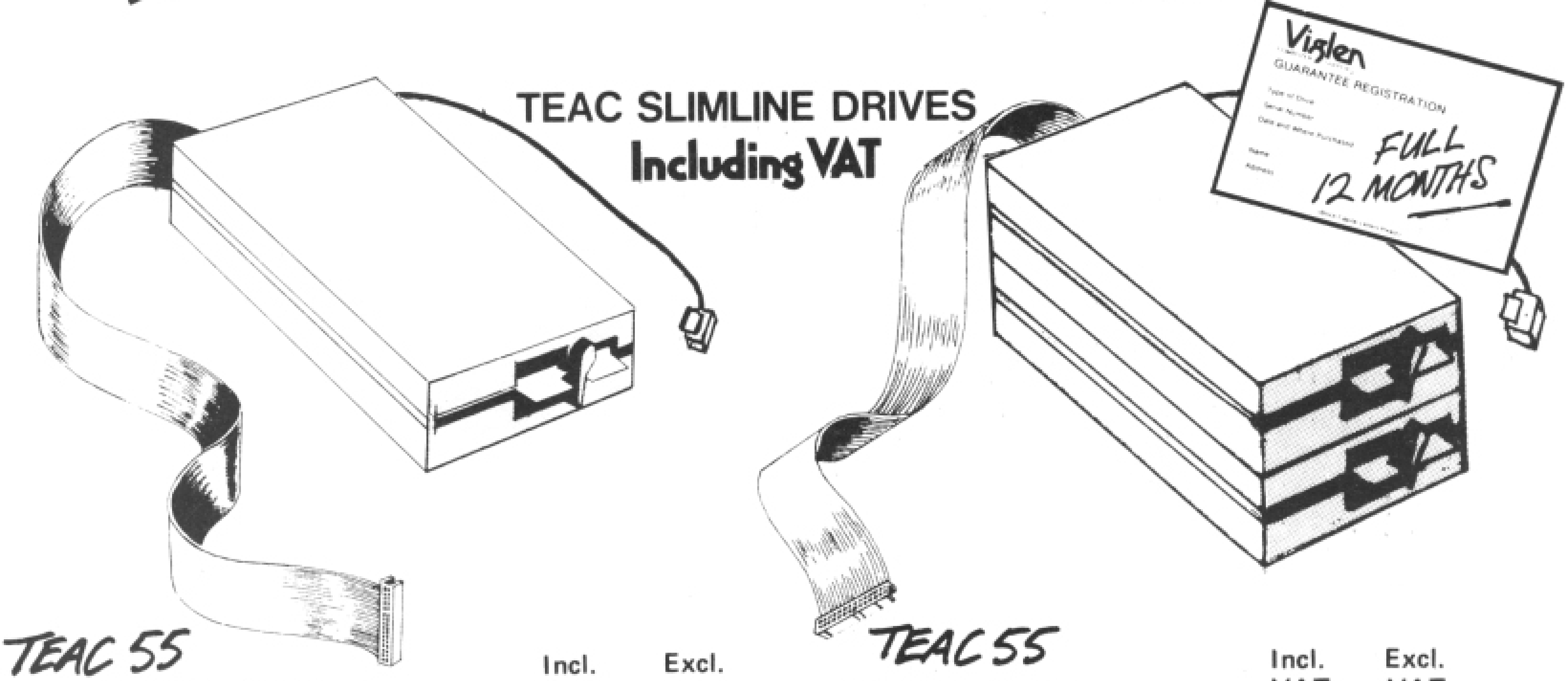
If you would rather not stay up all night typing in the program, I would be happy to oblige readers with copies of the program on cassette. The cost will be £3 which includes cassette, packaging and postage or £2 if you forward a blank cassette with the postage. Write to N Browne, 19 Kipling Road, Hilsea, Portsmouth, Hants.■

YOU CAN ALSO DO IT IN 5 SECONDS!

Replacing your Cassette with a Disk Drive, means DATA ACCESS in less than 5 seconds!

If you are looking for a guaranteed Disk Drive, fully BBC compatable, at a price that includes VAT, formating disk, user manual, cases and leads, then just look at our all-inclusive prices!

TEAC SLIMLINE DRIVES
Including VAT



TEAC 55

SINGLE DRIVES CASED	Incl. VAT	Excl. VAT
40 TRACK 100K	£166	£144.35
40 TRACK 200K	£230	£200.00
40/80 TRACK SWITCHABLE 200K	£199	£173.05
40/80 TRACK SWITCHABLE 400K	£257	£223.48

TEAC 55

DUAL DRIVES CASED	Incl. VAT	Excl. VAT
40 TRACK 200K	£320	£278.26
40/80 TRACK SWITCHABLE 400K	£402	£349.57
40/80 TRACK SWITCHABLE 800K	£520	£452.18
DFS KIT	£75	£65.22
P.S.U.	£32.20	£28

MITSUBISHI DRIVES	Excl. VAT	Incl. VAT
400K SINGLE DRIVE	£213.04	£245
800K DUAL	£379	£435.85
800K & POWER SUPPLY	£399	£458.85

*These drives are fully compatable with other Computers and can be used should you change your computer.

*Check our special terms for Educational Establishments and Government Departments

DEALER ENQUIRIES WELCOME

SPECIAL OFFER!
FULL HEIGHT DRIVE
£145 inc. VAT
READY TO PLUG IN

HOW TO ORDER

BY POST

To purchase any of the items simply fill in your requirements listed above, in the coupon. Enclose your cheque/P.O. or use your Access or Barclaycard. Please make cheque payable to: VIGLEN COMPUTER SUPPLIES and post to address below, allow 7 to 14 days for delivery & add £ 8—00 postage, package & insurance on all prices.

BY TELEPHONE
-(01) 843 9903
Credit Card holders (Access/Barclay Card only) can purchase by telephone. Please give Card No, Name, Address and the item required.

ENQUIRE ABOUT OUR 48-HOUR GUARANTEED DELIVERY SERVICE!

Post to; VIGLEN COMPUTER SUPPLIES UNIT 7 TRUMPERS WAY · HANWELL · W7 2QA

Please send me the following items : ____________

I enclose Cheque/P.O. for £ ____________

I prefer to pay Access/Barclaycard
(Delete whichever not applicable)

CARD No ____________
SIGNATURE ____________
NAME ____________
ADDRESS ____________

Tel (Day) ________ Tel (Eve) ________

CREDIT CARDS VALID IF SIGNED BY CARD HOLDER. ADDRESS ABOVE MUST BE THE SAME AS CARD HOLDER. DU 2/84

Viglen
COMPUTER SUPPLIES UNIT 7 TRUMPERS WAY
HANWELL W7 2QA
Tel: (01) 843 9903

EMS


Games that come from...
BEYOND
CHALLENGING SOFTWARE
UP PERISCOPE
PROTECT the convoy using SONAR + Depth charges to seek out and destroy the enemy below!
Try and pick off your Enemy from below the waves!
Kriegspiel
...How much longer can we hold this town...?
Here comes the first flakes of snow and out of it – their Tanks!
KRIEGSPIEL: A thrilling game of strategy to be played against the Dragon or any other devious apponent.

Easy animation

Dave Windle introduces the basics of animation for the Dragon and shows just how easy it can be

MOST OF THE home micros available now are capable of some form of animation, the Dragon 32 being one of the leaders in the field. On all micros there are various ways of achieving animation, although the Dragon has more than most. In this article we will concentrate on two methods of obtaining movement.

The simplest way

The first of these is the simplest way possible — using the text screen. Although written for Dragon 32 machines, this set of programs should run on most computers, with only slight changes.

Let us start by displaying on screen the "star" of our epic.

```
10 CLS
20 Y=0
30 PRINT@Y,"*"
```

If we now add the lines 40 Y=Y+1 and 50 GOTO 30 we have a line that flashes across the screen. By adding a space *before* the '*' in line 30 we can blank out the last '*' and give the illusion of movement (line 30 should now read 30 PRINT @ Y," *"). All this happens rather quickly, so let us rewrite our program to slow things down.

```
10 CLS
20 FOR Y=1 TO 30
30 PRINT@Y," *"
35 FOR K=1 TO 100:NEXT K
40 NEXT Y
```

The star's comeback

Our next problem is to bring our '*' back again. This can be done with the lines:

```
50 FOR Y=30 TO 1 STEP-1
60 PRINT @ Y,"*"
70 NEXT Y
80 GOTO 80
```

The program, as it stands, deals fairly well with movement in the horizontal plane, so how about vertical movement? This is quite possible on the TEXT.screen, if slightly more complicated. Using our original program, altered to read:

```
10 CLS
20 FOR Y=448 TO 0 STEP-32
30 PRINT@Y,"*"
40 FOR K=1 TO 100:NEXT K
50 NEXT Y: GOTO 10
```

We have lift-off. However this time, our erasing SPACE will not work. So we have to find another means of removing the last '*'. Using CLS will work, of course. Try changing line 50 to: 50 CLS: NEXT and we now have vertical movement. Using CLS is not much good, though, if you have anything else on the screen that you need to keep. Add the lines

```
5 CLS: PRINT @ 202,"gone"
7 FOR K=1 TO 500: NEXT
35 PRINT @ 224,"BLINK"
```

for a demonstration of some of the problems. We need then to find another way of erasing our '*'. What we need to do is print a space immediately below the NEXT print position. In other words in the space occupied by our '*' *before* the current cycle of the loop.

To do this we need to alter our program once more:

```
10 CLS
20 FOR Y=448 TO 0 STEP-32
30 PRINT@Y,"*"
40 FOR K=1 TO 100:NEXT K
50 PRINT @Y," "
60 FOR K=1 TO 100:NEXT K
70 NEXT Y
```

Now we have achieved movement in both planes. Let us now finally rewrite the program to demonstrate a bit more graphically what we have learned.

```
10 CLS0:PRINT @ 234,"MIDDLE";
20 FOR Y=448 TO 0 STEP-32
30 PRINT @ Y,"*";: GOSUB 90
40 PRINT @ Y," ";: GOSUB 90
50 NEXT Y
60 FOR Y=0 TO 30
70 PRINT @ Y," *";:GOSUB 90
80 NEXT Y: GOTO20
90 FOR K=1 TO 100:NEXT:RETURN
```

As you can see from running the program it is not finished. Using the information contained in the earlier listings see if you can complete the movement around the screen.

Method No. 2

Let us now look at another method of animation. This time we will use Dragon's excellent DRAW command.

The following program DRAWs a figure on the screen, clears it and then DRAWs a slightly different figure a few points forward. The program is quite simple and the following notes will help you to understand its workings.

Program notes

10-30	Sets MODE and SCREEN.
40-60	Creates STRINGS TO DRAW figures.
80-310	Makes SOUND and controls DRAWings.
320	Loops back to start. ■

```
5 '**WALKMAN**DAVE*WINDLE*AUG*83
10 PMODE4,1:SCREEN1,1:PCLS:DIM R
OC(29,39)
20 DIMBLA(29,39)             30 Y=120
40 R$="S6U9R2D9R2U9L2U2R2L4U2L1R
1U2R4D4BL2BD3G5E5D14L3BU6BR3F6L3"
50 L$="S6U9R2D9R2U9L2U2R2L4U2L1R
1U2R4D4BL2D3F5H5D14L3BU6BR3G6L3"
60 M$="U9R2D9R2U9L2U2R2L4U2L1R1U
2R4D4BL2BD11G6L3R3E6F6L3"
70 DRAW"BM220,90"+ L$
80 SOUND5,1
90 FOR K=1 TO 200:NEXT
100 PCLS
110 DRAW"BM200,90"+M$
115 SOUND50,1
120 FORK=1 TO 200:NEXT
130 PCLS
140 DRAW"BM180,90"+R$
145 SOUND5,1
150 FOR K=1 TO 200:NEXT
160 PCLS
170 DRAW"BM160,90"+L$
175 SOUND50,1
190 FORL=1 TO 200:NEXT:PCLS
200 DRAW"BM140,90"+M$
205 SOUND5,1
210 FORK=1TO200:NEXT:PCLS
220 DRAW"BM120,90"+R$
225 SOUND50,1
230 FORK=1TO200:NEXT:PCLS
240 DRAW"BM100,90"+L$
245 SOUND5,1
250 FORK=1TO200:NEXT:PCLS
260 DRAW"BM80,90"+M$
265 SOUND50,1
270 FORK=1TO200:NEXT:PCLS
280 DRAW"BM60,90"+R$
285 SOUND5,1
290 FORK=1TO200:NEXT:PCLS
300 DRAW"BM40,90"+L$
305 SOUND50,1
310 FORK=1TO200:NEXT:PCLS
320 GOTO60
```

Getting down to small business

*There's more to the micro than playing games — **Margaret Norman** explains some of the methods and applications of data storage and manipulation*

MOST HOME computer owners bought their micros in the belief that they would be able not only to play games on them, but also use them to keep track of their bank accounts, look after their files and so on. Here is a program which I hope will make it easier for you to do some of these things, by showing you how to store and manipulate data.

This program has been written to hold the names of a number of items, further brief descriptions (eg the category to which each belongs) and an associated numeric value. It can calculate the total numeric value of various numbers of different items for you. There are several possible applications for a program like this: it could be used to calculate the value of stock in a small shop, by entering the names of items stocked and their prices, or even to calculate the calorie content of a meal, by entering names of foods and their calorie values.

File structure

It could also be used just as a simple file for, say, names, addresses and telephone numbers — you can just ignore the part of the program which performs the calculations, or remove it by deleting lines 260 and 5000 to 5150, changing the number 5 in lines 270 to 290 to 4, and omitting the number 5000 from line 300.

The first stage in writing such a program is to determine the structure of the file to be used, the number and type of data fields and the maximum number of records that can be handled. The number of records which can be held in memory at once obviously depends on the number of fields in each: the more fields, the less records you can have. The data is held in arrays; here a string array is used for the first two fields, and a numeric array for the third. It would be possible to hold all the data in a string array, using the VAL function where calculations are to be performed, but if it is known that calculations will be required it is more efficient to use one numeric array.

Alphabetical order

If you are unsure how many records your file will hold, you can find out by trial and error; put a fairly large number in the DIM statement at the start of the program, then keep entering data until you get an OM error. PRINT N will then give you the number of records it took. If you do this, remember to record the data at regular intervals so you don't lose it all.

The records are arranged in alphabetical order by the sort routine in lines 2050 to 2120. A very simple sort has been used, which will place records in the file fastest if they are entered already in alphabetical order. For data which is likely to be entered in a random order, a more complex sort routine, such as a binary sort, would be quicker. It would be a simple task to change the routine so that records are filed according to the numeric value rather than the name of the item — just change every occurrence of A$(X,1) in this section to A(X).

Deciding options

Once you have structured the file, the next job is to decide which options you want to have available to the program user. Obviously you will need to be able to enter new data, to save the data on tape and to load it again (there is little point in having a file if you cannot also examine entries). An option to delete entries is also fairly essential — here it has been combined with the examine data option. If your records are fairly long ones it may be desirable to include an option to alter the data in individual fields — here, changes can only be made by deleting then re-entering a complete record. You may also want to be able to search for all the records containing a given string; use the INSTR function for this.

The option to load an existing file from tape is given at the beginning of the program, as this can only be done at the start. All other options are presented in ▶

```
10 REM ADDFILE
20 REM BY MARGARET NORMAN
30 PCLEAR 1:CLEAR 10000
40 DIM A$(400,2),A(400)
50 CLS4
60 PRINT@204,"ADDFILE";
70 FOR DL=0 TO 1000:NEXT
100 CLS
110 INPUT"DO YOU WISH TO LOAD EXISTING     DATA FILE FROM TAPE";Q$
120 IF Q$="Y" THEN 1010
200 REM MAIN OPTIONS
210 CLS
220 PRINT@37,"OPTIONS AVAILABLE:":PRINT
230 PRINT"(1) ADD NEW DATA TO FILE"
240 PRINT"(2) EXAMINE/DELETE DATA IN FILE"
250 PRINT"(3) SAVE FILE ON TAPE"
260 PRINT"(4) CALCULATION"
270 PRINT"(5) STOP"
280 PRINT:INPUT"ENTER NO. BETWEEN 1 AND 5";OPT
290 IF OPT<1 OR OPT>5 THEN 280
300 ON OPT GOTO 2000,3000,4000,5000,350
350 CLS:PRINT@268,"GOODBYE":END
1000 REM LOAD FILE FROM TAPE
1010 CLS:PRINT"WHEN TAPE IS READY,PRESS'ENTER'";:INPUT Q$
```

Deltas for Dragons



Complete control at your fingertips.

The smooth control of a Voltmace DELTA sprung return joystick is now available to Dragon owners. Called the DELTA 3D it is a cut down version of our DELTA 14B handset.
The same robust ball and socket jointed joystick but with the keypad blanked off to leave a choice of one of three fire buttons.

- Nylon encased – Steel shafted joysticks with ball and socket joint
- Fast spring return to centre
- Graphite wiper linear potentiometers

DELTA 3D joystick .... £10.00 each
or £19.50 per pair

Voltmace Ltd
PARK DRIVE, BALDOCK, HERTS. SG7 6E
Tel: (0462) 894410
Callers welcome at the factory – Monday to Friday
Made in England. Trade terms available.

Voltmace delta 3d

---

advanced sound & graphics
for the Dragon computer

including machine code subroutines

keith & steven brain



Dragon book from Sunshine

Advanced Sound & Graphics for the Dragon computer
by Keith and Steven Brain £5.95

Advanced Sound & Graphics uses a carefully structured approach to show you how to develop routines in your own Dragon programs.
All the major aspects of the sound and graphics capabilities are covered in detail and are fully illustrated. The book takes you from first principles through to bar charts, maps, 3-D projections, movement, animation, direct drawing, screen saving and printing and many other features.
Complex sound effects are examined in detail including keyboard sound synthesis, the graphic display of music and the integration of sound and graphics.
In addition to dealing with the operation and applications of the BASIC commands the book explains the internal organisation of the sound and graphics facilities. It also shows you how to use machine code routines to improve your programs.
Keith and Steven Brain have already published the best selling book *Dragon 32 Games Master.*

Sunshine
Please send me .... copies of *Advanced Sound & Graphics* at £5.95 each.

I enclose a cheque/postal order for £. . . . .
payable to Sunshine, 12-13 Little Newport Street, London WC2R 3LD.

Name ....

Address...

....

Signed...

or ring 01-734 3454 to order by Access/Mastercard

We can normally deliver in four to five days
Sunshine — publishers of *Dragon User, Popular Computing Weekly,* books and software

```
1020 OPEN"I",£-1,"ADDF"
1030 INPUT £-1,N
1040 FOR I=1 TO N
1050 INPUT £-1,A$(I,1)
1060 INPUT £-1,A$(I,2)
1070 INPUT £-1,A(I):NEXT I
1080 GOTO 200
2000 REM ADD DATA TO FILE
2010 IF N=400 THEN CLS:PRINT@265,"FILE IS FULL":FOR DL=0 TO 1000:NEXT:GOTO 200
2020 CLS:INPUT"NAME OF ITEM";A$(N+1,1)
2030 INPUT "DESCRIPTION";A$(N+1,2)
2040 INPUT"VALUE";A(N+1)
2050 IF N=0 THEN 2120
2060 FOR I=N TO 1 STEP -1
2070 IF A$(I+1,1)>=A$(I,1) THEN 2120
2080 T$(1)=A$(I+1,1):T$(2)=A$(I+1,2):T=A(I+1)
2090 A$(I+1,1)=A$(I,1):A$(I+1,2)=A$(I,2):A(I+1)=A(I)
2100 A$(I,1)=T$(1):A$(I,2)=T$(2):A(I)=T
2110 NEXT I
2120 N=N+1
2130 CLS:INPUT"ANY MORE NEW DATA (Y/N)";Q$
2140 IF Q$="Y" THEN 2010 ELSE 200
3000 REM EXAMINE/DELETE DATA
3010 CLS
3020 PRINT"DO YOU WANT:"
3030 PRINT"(1) THE WHOLE FILE"
3040 PRINT"(2) A PARTICULAR ENTRY"
3050 PRINT:INPUT"ENTER 1 OR 2";X
3060 IF X<>1 AND X<>2 THEN 3050
3070 IF X=1 THEN I=1:GOTO 3140
3080 REM FIND ENTRY
3090 CLS:INPUT"ENTER NAME";S$
3100 FOR I=1 TO N
3110 IF S$=A$(I,1) THEN 3140
3120 NEXT I
3130 CLS:PRINT"THIS ITEM IS NOT IN FILE":FOR DL=0 TO 1000:NEXT:GOTO 200
3140 CLS:PRINTA$(I,1):PRINT A$(I,2):PRINT "VALUE: ";A(I)
3150 PRINT:PRINT"DO YOU WANT TO:"
3160 PRINT"(1) DELETE THIS ENTRY"
3170 PRINT"(2) EXAMINE NEXT ENTRY"
3180 PRINT"(3) SELECT NEW OPTION"
3190 PRINT:INPUT"ENTER 1,2 OR 3";Y
3200 IF Y<1 OR Y>3 THEN 3190
3210 ON Y GOTO 3240,3220,200
3220 IF I<N THEN I=I+1:GOTO 3140
3230 CLS:PRINT@265,"END OF FILE":FOR DL=0 TO 1000:NEXT:GOTO 200
3240 REM DELETE ENTRY
3250 IF I=N THEN 3280
3260 FOR J=I TO (N-1)
3270 A$(J,1)=A$(J+1,1):A$(J,2)=A$(J+1,2):A(J)=A(J+1):NEXT J
3280 N=N-1
3290 CLS:PRINT"ENTRY DELETED":PRINT
3300 PRINT"DO YOU WANT TO:"
3310 PRINT"(1) EXAMINE NEXT ENTRY"
3320 PRINT"(2) SELECT NEW OPTION"
3330 PRINT:INPUT"ENTER 1 OR 2";Z
```

◀ a menu, to which the program returns when each task has been completed.

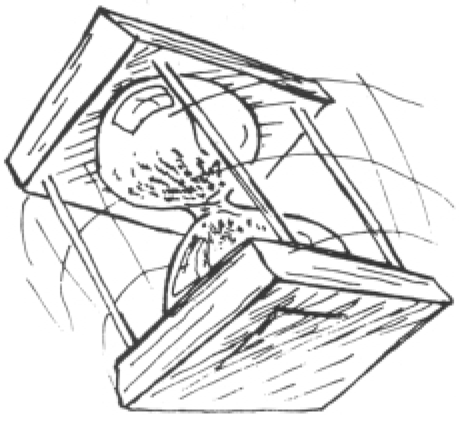
The routines which handle the saving and loading of data on tape have been simplified by making the first entry in the tape file the number of records in the file; this means there is no need to use an end-of-file marker. The program pauses at the start of these routines (waits for ENTER to be pressed) to give the user time to position the tape and put the recorder in the correct mode. You could if you wish insert a MOTOR ON command to facilitate the positioning of the tape.

The section of the program which performs the calculations is also very simple. You are asked for the name of an item; the appropriate file entry is found, then the description and value are printed and you are asked for the number of these items. The computer then calculates the value of this number of items and gives you this figure and a running total. If the name you have entered is not in the file you are informed of this and asked for another. This is important as the filed names are only checked for an exact match with the word you have entered, so if you spell the name differently the appropriate entry will not be found.

All programs involving the use of data files should be crashproofed as well as possible, to minimise the risk of data being lost. Every request for input should be accompanied by a clear indication of the form in which it is required, and followed by a check to see that it does fall within the required range, especially if it is to be used in an ON . . . GOTO statement. There are plenty of examples of how to do this, eg lines 280-290 check the selection of an option from the main menu. All selections of options in this program are checked in this way, but no checks are made here on the actual data in the files. If you are writing, say, a financial program where typing an extra zero by mistake could prove costly, it is obviously worth ▶

SUPERIOR SO

EDIT+



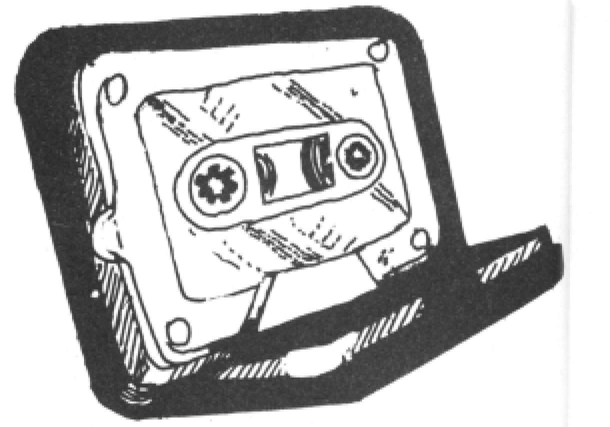
* EDIT+ is a Full Screen Editor and Programmer's Tool Kit. It's an excellent aid for writing programs in BASIC and is easy to use for the novice as well as the experienced programmer. EDIT+ includes all the facilities of HI-RES. Up to 23 lines of your program are displayed on the screen and can be changed by overtyping, inserting, or deleting characters. Functions include: Find String, Change String, Copy Text, Goto Specified Line, Scroll Up Down, Append From Tape and Enter Basic Command. No Dragon is complete without an EDIT+. £34.50

HI~RES



* Plug the HI-RES cartridge into your DRAGON and you will immediately see the improvement. The screen displays 24 rows of 51 characters with proper lower case and BASIC works as normal but with extra features: Selectable character sets (English, French, German, Danish, Swedish, Italian, Spanish). SPRITE Graphics. Redefinable Characters. Improved keyboard action with autorepeat allows faster typing. Graphics and text can be mixed on the screen. Suitable for educational and business use. £25.30

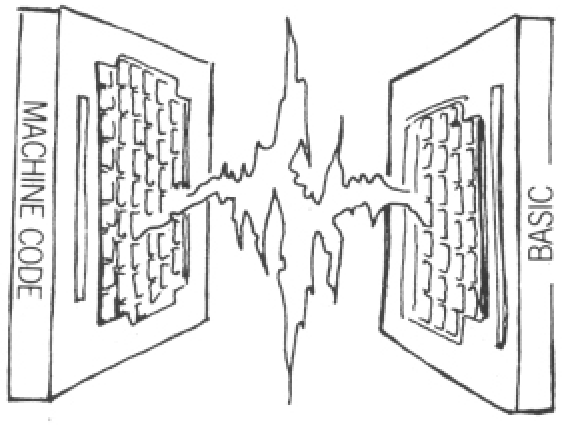
SOURCE TAPES



The following programmes contain both source and object code. They can be used in conjunction with 'DASM' or on their own as individual programmes. It's a great way to build up your software library. Each tape represents excellent value at only £5.99 each.

1. DISASSEMBLER
2. GAME OF LIFE
3. HI-RES SCREEN DUMP FOR EPSOM
4. HI-RES SCREEN DUMP FOR SEIKOSHA

DASM



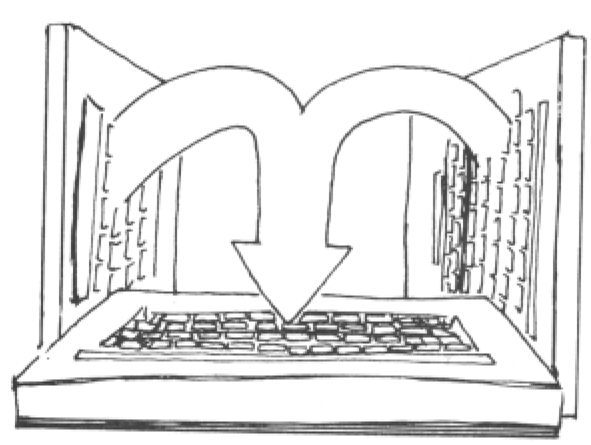
DASM is a versatile assembler, designed especially for ease of use on the DRAGON and allows you to assemble machine code while still retaining the full use of BASIC. Supports all 6809 instructions and modes. Allows any length for labels (the first 5 and the last characters are used). Full support for output to printer. Recommended for the beginner. £18.95

DEMON



A powerful machine code monitor which allows you to delve into the internals of your DRAGON as well as helping you to debug your machine code programs (and BASIC programs using PEEK and POKE). Includes: Examine/Change memory, Examine/Change registers, Print Screen, Set Breakpoints, Test Memory. An essential tool for all machine code users. £18.95

DASM/DEMON



* It has all the features of both DASM and DEMON in one package. DEMON is the natural-partner to DASM, complementing each other perfectly. Write, test and use your programmes without the bother of reloading. It is extensively featured in the new book by Ian Sinclair on Dragon Machine Code. It is the ideal combination for the machine code user. £30.45

VISIT YOUR NEAREST STOCKIST OR FOR FAST MAIL ORDER SERVICE C

```
3340 IF Z<>1 AND Z<>2 THEN 3330
3350 ON Z GOTO 3220,200
4000 REM SAVE FILE ON TAPE
4010 CLS:PRINT"WHEN TAPE IS READY,PRESS'ENTER'";:INPUT Q$
4020 OPEN"O",£-1,"ADDF"
4030 PRINT £-1,N
4040 FOR I=1 TO N
4050 PRINT £-1,A$(I,1):PRINT £-1,A$(I,2)
4060 PRINT £-1,A(I):NEXT I
4070 CLOSE £-1
4080 GOTO 200    5000 REM CALCULATION
5005 T=0:CLS
5010 INPUT"NAME";F$
5020 FOR I=1 TO N
5030 IF F$=A$(I,1) THEN 5100
5040 NEXT I
5050 PRINT"THIS ITEM IS NOT IN FILE":GOTO 5130
5100 PRINTA$(I,2):PRINT"VALUE: ";A(I)
5110 INPUT"NUMBER";M
5120 PRINT"TOTAL VALUE:";M*A(I):T =T+M*A(I):PRINT "RUNNING TOTAL:";T
5130 PRINT:INPUT"ANY MORE ITEMS (Y/N)";R$
5140 IF R$="Y" THEN 5010        5150 GOTO 200
```

◀ checking that data lies within certain limits or asking for additional confirmation of very large items.

A short program like this, adapted to suit your needs, will deal with a wide range of minor tasks but obviously it has its limitations. If your requirements are much more complex then it is well worth looking at some of the business programs on the market. A clear understanding of how this program operates, though, should make it much easier to get to grips with its big brothers. ■

Program structure

Lines	
10-70	initialisation, title
100-120	load file option
200-300	main options menu
350	end
1000-1080	load file from tape
2000-2140	enter new data
3000-3230	examine data; delete data option
3240-3350	delete data
4000-4080	save file on tape
5000-5150	calculation

Variables used

A$(I,1)	name of item I
A$(I,2)	description of item I
A(I)	value of item I
DL	delay loop counter
N	no. of records on file
Q$,R$	Y/N inputs
OPT,X,Y,Z	options selected
I,J	loop counters
S$	search string (name of item)
T	total (in calculation)

Recovering from i/o errors

***Pam D'Arcy** shows how to sort out your tapes with Tapescan*

IN PRACTICALLY EVERY issue of any computer magazine we commonly read complaints about the lack of a verify facility on the Dragon. I have not worried too much about this as there are means of overcoming it. The lack of a tape catalogue facility irritates me although, again, SKIPF can be used to check that the tape content matches one's records.

My pet outrage is the way that everything grinds to a halt on detection of an I-O error. The biggest drawback for me is that this thwarts most attempts at cutting down on loading time by fast-winding tapes to an approximate position. Brief encounters on a BBC machine show that all blocks within their files are numbered, the blocks are identified as they are passed/loaded and, having been informed of an I-O error, you are allowed to reposition the tape to attempt a re-read of the problem block if it occurred during loading.

The Dragon gives you no idea of how far you are through the current file, be it one being loaded or skipped, so my usual trick of "SKIPF" until the tape reaches the start of the next file may need to be typed in only a few times — or so many times that I give up and rewind the whole tape back and start again anyway!

Having received the Dragon Data "Information for Machine Code Users" leaflet (and armed with tape layout information in CoCo's Technical Reference Manual), I set about writing a tape listing program with two main criteria:

1. Recovery from I-O errors to allow faster tape positioning.
2. It should be machine coded so that it could be permanently resident in the Dragon and be used without affecting any other program currently loaded (notably, Basic!).

The resulting machine code listing, Tapescan, is published with this article. It doesn't assist with CLOAD/SKIPF and program read/write errors in that it is not intercepting any of those routines — but it *is* there to assist with checking the tape(s) alongside other programs when you may be experiencing problems.

Tapescan can be entered directly using the TOPSY program featured in the June issue. My initial entry took only about 20 minutes, including checking out my own coding queries.

Or you can use the Sets can loader which is also published here.

Brief details of its use and technical details now follow. Information on the tape file formats can be ascertained from the listing.

Between files, the cassette motor is switched off for approximately three seconds. If no key is pressed in this time, the run continues; if Q is pressed, the run is terminated. Pressing any other key leaves the motor switched off until another key is pressed (Q will again quit the program), enabling tapes to be changed or the screen display to be studied.

If a read error is detected, the program resumes by hunting for a file header block. As data blocks containing binary information (eg CSAVE, CSAVEM files) are stored without inter-block gaps, obviously, depending on where within a file the ►

```
1 REM SETSCAN   (C)PAM D'ARCY
2 REM JUNE 1983
3 CLEAR200,32093:MADDR=32093:MBY
TES=413
4 CLS:PRINT"SETSCAN PROGRAM"
5 PRINT:PRINT"SETSCAN LOADS THE
TAPESCAN        MACHINE CODE PROGR
AM WHICH IS     STORED NEXT ON THI
S TAPE THEN     ALLOWS YOU TO COPY
 BOTH PROGRAMSON TO ANOTHER TAPE
"
6 PRINT:PRINT"BEFORE SAVING THE
PROGRAMS,       A MOTORON OPTION I
S AVAILABLE     TO ENABLE THE RECE
IVING TAPE TO BE POSTIONED CORRE
CTLY BEFORE     COMMENCING SAVES"
7 FOR N=1 TO 7000:NEXT
8 PRINT:PRINT"LOADING TAPESCAN"
9 CLOADM"TAPESCAN"
10 PRINT:PRINT"TAPESCAN LOADED"
11 CLS:PRINT"SETSCAN PROGRAM"
12 PRINT
13 PRINT"PRESS Q TO QUIT; M TO M
OTORON    (TO POSITION TAPE PRIOR
 TO        SAVING PROGRAMS); ELSE
ANY OTHERTO COMMENCE SAVES"
14 K$=INKEY$:IFK$=""THEN14
15 IFK$="Q" THEN CLS:END
16 IFK$<>"M" THEN 22
17 MOTORON
18 PRINT:PRINT"MOTOR IS NOW ON":
PRINT:PRINT"PRESS Q TO QUIT PROG
RAM;          ELSE ANY OTHER KEY T
O HALT MOTOR"
19 K$=INKEY$:IFK$="" THEN 19
20 MOTOROFF
21 IFK$="Q" THEN CLS:END
22 CLS:PRINT"SETSCAN PROGRAM"
23 PRINT:PRINT"PRESS Q TO QUIT E
LSE PUT          RECORDER INTO 'RE
CORD' MODE AND PRESS ANY OTHER T
O COMMENCE       SAVES"
24 K$=INKEY$:IF K$=""THEN24
25 IFK$="Q" THEN CLS:END
26 CLS:PRINT "SETSCAN PROGRAM":P
RINT:PRINT"SAVING PROGRAMS NOW"
27 CSAVE"SETSCAN"
28 CSAVEM"TAPESCAN",MADDR,MADDR+
(MBYTES-1),MADDR
29 PRINT:PRINT"PROGRAMS JUST SAV
ED"
30 PRINT:PRINT"TAKE RECORDER OUT
 OF 'RECORD'    MODE"
31 PRINT:PRINT"VERIFICATION OF S
AVE REQUIRED?  (QUIT PROGRAM Q;
VERIFY V;        ELSE ANY)"
32 K$=INKEY$:IF K$="" THEN 32
33 IFK$="Q" THEN CLS:END
34 IFK$<>"V" THEN 11
35 CLS:PRINT"SETSCAN PROGRAM"
36 PRINT:PRINT"VERIFICATION OF S
AVES REQUESTED"
37 PRINT:PRINT"TAPESCAN WILL NOW
 BE ENTERED;    POSITION TAPE WHE
N YOU HEAR THE MOTOR COME ON;
                PRESS Q AT THEN C
ONCLUSION OF    THE REQUIRED VERI
FICATION TO     QUIT TAPESCAN PRO
GRAM"
38 FOR N=1 TO 7000:NEXT
39 EXEC
40 PRINT:PRINT"BACK IN SETSCAN;
PRESS Q TO QUITOR ANY OTHER TO C
ONTINUE"
41 K$=INKEY$:IFK$=""THEN41
42 IFK$="Q"THEN CLS:END
43 GOTO11
```

Setscan loader – for loading Tapescan

```
Address  Object code    Op
7D5D 20 09              BRA    >7D68
Preset data area, title
7D5F 54 41 50 45        FCC    "TAPESCAN"
     53 43 41 4E
7D67 8D                 FCB    $8D
Program, clear screen
7D68 86 60              LDA    #$60
7D6A 8E 04 00           LDX    #$0400
7D6D 9F 88              STX    $88
7D6F A7 80              STA    ,X+
7D71 8C 06 00           CMPX   #$0600
7D74 26 F9              BNE    >7D6F
Screen cleared, display title
7D76 30 8C E6           LEAX   >7D5F,PC
7D79 A6 80              LDA    ,X+
7D7B 1F 89              TFR    A,B
7D7D 84 7F              ANDA   #$7F
7D7F BD 80 0C           JSR    $800C
7D82 C4 80              ANDB   #$80
7D84 27 F3              BEQ    >7D79
Titling complete
7D86 30 8D 01 77        LEAX   >7F01,PC
7D8A 9F 7E              STX    $7E
7D8C 20 20              BRA    >7DAE
Motoroff pause between files of about 3
seconds to allow user to halt
7D8E 8E 60 00           LDX    #$6000
7D91 BD 80 06           JSR    $8006
7D94 81 00              CMPA   #$00
7D96 26 06              BNE    >7D9E
7D98 30 1F              LEAX   -01,X
7D9A 26 F5              BNE    >7D91
7D9C 20 10              BRA    >7DAE
Key pressed – halt
7D9E 81 51              CMPA   #$51
```

```
7DA0 26 01              BNE    >7DA3
Exit from program requested
7DA2 39                 RTS
Wait for another key to be pressed before
proceeding
7DA3 BD 80 06           JSR    $8006
7DA6 81 00              CMPA   #$00
7DA8 27 F9              BEQ    >7DA3
Key pressed
7DAA 81 51              CMPA   #$51
7DAC 27 F4              BEQ    >7DA2
Get first block of a new run or expected file
header block (previous= EOF) or recover
after read error
7DAE BD 80 21           JSR    $8021
7DB1 AD 9F A0 06        JSR    [$A006]
7DB5 D6 81              LDB    $81
7DB7 27 04              BEQ    >7DBD
Read error occurred
7DB9 86 78              LDA    #$78
7DBB 20 06              BRA    >7DC3
Error free read
7DBD D6 7C              LDB    $7C
7DBF 27 18              BEQ    >7DD9
Not a name block
7DC1 86 6E              LDA    #$6E
Output the error message
7DC3 34 06              PSHS   B,A
7DC5 BD 80 18           JSR    $8018
7DC8 86 0D              LDA    #$0D
7DCA BD 80 0C           JSR    $800C
7DCD 35 02              PULS   A
7DCF BD 80 0C           JSR    $800C
7DD2 35 02              PULS   A
7DD4 17 01 03           LBSR   >7EDA
7DD7 20 B5              BRA    >7D8E
Process namefile block
7DD9 86 0D              LDA    #$0D
7DDB BD 80 0C           JSR    $800C
Display name of file
```

```
7DDE 9E 7E              LDX    $7E
7DE0 C6 08              LDB    #$08
7DE2 A6 80              LDA    ,X+
7DE4 BD 80 0C           JSR    $800C
7DE7 5A                 DECB
7DE8 26 F8              BNE    >7DE2
Display file type (00=Basic, 01=data,
02=m/c)
7DEA 86 20              LDA    #$20
7DEC BD 80 0C           JSR    $800C
7DEF A6 84              LDA    00,X
7DF1 A7 8D 01 0B        STA    >7F00,PC
Convert it to an alpha character
7DF5 8B 42              ADDA   #$42
7DF7 81 42              CMPA   #$42
7DF9 27 08              BEQ    >7E03
7DFB 8B 01              ADDA   #$01
7DFD 81 44              CMPA   #$44
7DFF 27 02              BEQ    >7E03
7E01 86 4D              LDA    #$4D
7E03 BD 80 0C           JSR    $800C
Display A if ASCII format+ Basic file (else
space char)
7E06 E6 01              LDB    01,X
7E08 C1 FF              CMPB   #$FF
7E0A 26 08              BNE    >7E14
7E0C 81 42              CMPA   #$42
7E0E 26 04              BNE    >7E14
7E10 86 41              LDA    #$41
7E12 20 02              BRA    >7E16
7E14 86 20              LDA    #$20
7E16 BD 80 0C           JSR    $800C
Save inter-block gap flag (00= continuous
data=binary, FF=gaps (with sync)=
ASCII/data)
7E19 A6 02              LDA    02,X
7E1B A7 8D 00 E0        STA    >7EFF,PC
Save m/c addresses in name block
7E1F EC 03              LDD    03,X
7E21 ED 8D 00 D7        STD    >7EFC,PC
```

◄ initial error occurred, more errors may be reported before the start of the next file is detected (the important thing is that the program keeps going!).

During the "motor on" time, you can of course take the recorder out of read mode and skip along the tapes forwards/backwards as desired.

The screen display is as follows: FILENAME — the eight-character name that the file was created with; TYPE — B (Basic), D (Data) or M (machine code) file type; ASCII FLAG — A if it is a Basic file saved in ASCII format (CSAVE . . .,A option); BLOCK COUNT — the count of the number of DATA blocks between the Header block and any EOF block (if any), displayed in hex; EOF FLAG — displays an upward arrow symbol if a separate EOF block is present; ADDRESSES (if machine code file) — load and start in hex.

Error codes (displayed in reverse screen) are:

x — Error occurred (it is followed by the error code returned by the Basic read subroutine displayed in hex; the only error mentioned in the grey Dragon Data leaflet is a checksum error on the block).

n — Not a Header block when expected (eg at the start of a run and will undoubtedly occur when reading past blocks following an I-O error); it is followed by the block type actually read, displayed in hex (01 Data block; FF EOF block).

h — Header block occurred before the previous file had been correctly terminated (ie EOF block missing, perhaps when a program had failed while writing a data file and the file was not closed).

Tapescan occupies 675 bytes of storage, including a 255-byte input buffer, thus the highest address that it can be loaded at is 32093 (hex 7D5D). That is why the addresses on the simulated machine code listing start at this address. This acts as a more than useful guide when entering the program using Topsy as if you give that address as the start address for ►

```
7E25 EC 05          LDD    05,X
7E27 ED 8D 00 CF    STD    >7EFA,PC
Clear and display block count
7E2B 86 00          LDA    #$00
7E2D A7 8D 00 CD    STA    >7EFE,PC
7E31 17 00 A6       LBSR   >7EDA
7E34 20 08          BRA    >7E3E
Check if sync needed (gap flag=FF)
7E36 A6 8D 00 C5    LDA    >7EFF,PC
7E3A 81 FF          CMPA   #$FF
7E3C 26 03          BNE    >7E41
7E3E BD 80 21       JSR    $8021
Get expected data or EOF block
7E41 AD 9F A0 06    JSR    [$A006]
7E45 96 81          LDA    $81
7E47 27 1C          BEQ    >7E65
Read error occurred
7E49 BD 80 18       JSR    $8018
7E4C 86 78          LDA    #$78
7E4E BD 80 0C       JSR    $800C
7E51 96 81          LDA    $81
7E53 17 00 84       LBSR   >7EDA
7E56 A6 AD 00 A6    LDA    >7F00,PC
7E5A 81 02          CMPA   #$02
7E5C 10 26 FF 2E    LBNE   >7D8E
M/c file was being read - output addresses
7E60 8D 62          BSR    >7EC4
7E62 16 FF 29       LBRA   >7D8E
Error free read
7E65 96 7C          LDA    $7C
7E67 26 14          BNE    >7E7D
File header block without previous file
seeming to be complete
7E69 86 68          LDA    #$68
7E6B BD 80 0C       JSR    $800C
7E6E A6 AD 00 8E    LDA    >7F00,PC
7E72 81 02          CMPA   #$02
7E74 10 26 FF 61    LBNE   >7DD9
Failed on m/c file - output m/c addresses
```

```
7E78 8D 3F          BSR    >7EB9
7E7A 16 FF 5C       LBRA   >7DD9
Valid data/EOF block
7E7D 81 FF          CMPA   #$FF
7E7F 27 22          BEQ    >7EA3
Data block - increment and display block
count
7E81 9E 88          LDX    $88
7E83 30 1E          LEAX   -02,X
7E85 9F 88          STX    $88
7E87 A6 8C 74       LDA    >7EFE,PC
7E8A 4C             INCA
7E8B A7 8C 70       STA    >7EFE,PC
7E8E 8D 4A          BSR    >7EDA
M/c files are at end if block length <255
7E90 A6 8C 6D       LDA    >7F00,PC
7E93 81 02          CMPA   #$02
7E95 10 26 FF 9D    LBNE   >7E36
7E99 96 7D          LDA    $7D
7E9B 81 FF          CMPA   #$FF
7E9D 27 A2          BEQ    >7E41
M/c file ended (no separate EOF block)
7E9F 86 20          LDA    #$20
7EA1 20 02          BRA    >7EA5
EOF block read (if m/c file length is integral
of 255 bytes, has an EOF block)
7EA3 86 5E          LDA    #$5E
7EA5 BD 80 0C       JSR    $800C
7EA8 BD 80 18       JSR    $8018
7EAB A6 8C 52       LDA    >7F00,PC
7EAE 81 02          CMPA   #$02
7EB0 10 26 FE DA    LBNE   >7D8E
M/c file - output 2 space chars+m/c address
from HDR block
7EB4 8D 03          BSR    >7EB9
7EB6 16 FE D5       LBRA   >7D8E
Subroutines, display 2 space chars+m/c
addresses from HDR block
7EB9 86 20          LDA    #$20
7EBB BD 80 0C       JSR    $800C
7EBE BD 80 0C       JSR    $800C
7EC1 8D 01          BSR    >7EC4
```

```
7EC3 39             RTS
Display m/c addresses in hex
7EC4 30 8C 33       LEAX   >7EFA,PC
7EC7 8D 03          BSR    >7ECC
7EC9 8D 01          BSR    >7ECC
7ECB 39             RTS
Display space char+2 bytes (addressed by
reg X) in hex
7ECC 86 20          LDA    #$20
7ECE BD 80 0C       JSR    $800C
7ED1 A6 80          LDA    ,X+
7ED3 8D 05          BSR    >7EDA
7ED5 A6 80          LDA    ,X+
7ED7 8D 01          BSR    >7EDA
7ED9 39             RTS
Display char in reg A in hex
7EDA 34 04          PSHS   B
7EDC 1F 89          TFR    A,B
7EDE 44             LSRA
7EDF 44             LSRA
7EE0 44             LSRA
7EE1 44             LSRA
7EE2 12             NOP
7EE3 8D 09          BSR    >7EEE
7EE5 1F 98          TFR    B,A
7EE7 84 0F          ANDA   #$0F
7EE9 8D 03          BSR    >7EEE
7EEB 35 04          PULS   B
7EED 39             RTS
Convert value in reg A (O-F) to ASCII char
and display
7EEE 8B 30          ADDA   #$30
7EF0 81 3A          CMPA   #$3A
7EF2 25 02          BCS    >7EF6
Alpha A-F
7EF4 8B 07          ADDA   #$07
7EF6 BD 80 0C       JSR    $800C
7EF9 39             RTS
Variable data area
```

◀ writing hex strings, the "hex write prompt" will correspond exactly with the line of coding to be entered.

Once saved (start address 32093, length 413, entry point 0) — and the save verified with the program itself! — the program may be loaded at any time with CLOADM, ensuring first that from at least 32093 memory is reserved for machine code (CLEAR 200,32093). (Should you get — shall we say — an interesting screen background to the title, you will probably find that you have failed to reserve the machine code area prior to loading!)

I save a copy of Tapescan at the beginning of all my tapes for instant accessibility and load it into the Dragon every time I switch on — it saves an awful lot of hassle.

Only 413 bytes need to be saved as the remainder of the 675 bytes is the variable data area:

7EFA	RMB 2	DFMLOAD
7EFC	RMB 2	DFMENTRY
7EFE	RMB 1	DFBLKCNT
7EFF	RMB 1	DFGAP
7F00	RMB 1	DFTYPE
7F01	RMB 255	DBUFF

Even as I am writing this, my mind is working on further developments of Tapescan . . . How about ensuring that the processor is currently working at the slower speed before accessing the tape for the first time (*ST*ore any register to hex FFD6)? . . . Is diverting the display to a printer (should you be so lucky!): just a matter of altering the "JSR $800C" instructions to "JSR $800F" . . .?

Should you have found it too daunting a prospect to enter the Topsy code from the published listing, I am prepared to send readers a copy on cassette by return post on receipt of £3.50; for another £1, I will include Tapescan on the cassette, too. Address: Pam D'Arcy, 21 Wycombe Lane, Wooburn Green, High Wycombe, Bucks HP10 0HD. ■

The easy way to interfacing

*Follow **A G Nanson's** instructions to build an analogue/digital interface – next month we'll show you how to use the interface to turn your Dragon into a simple storage oscilloscope.*

TOP ROW		BOTTOM ROW	
Reading from Right to Left		Reading from Right to Left	
1	+ 12 VOLT	2	+ 12 VOLT
3	HALT	4	NMI
5	RESET	6	E IN
7	Q IN	8	CART
9	+ 5 VOLT	10	DO
11	D1	12	D2
13	D3	14	D4
15	D5	16	D6
17	D7	18	R/W
19	AO	20	A1
21	A2	22	A3
23	A4	24	A5
25	A6	26	A7
27	A8	28	A9
29	A10	30	A11
31	A12	32	R2
33	GROUND	34	GROUND
35	SND	36	P2
37	A13	38	A14
39	A15	40	EXT MEM

Diagram 1: showing the pin out of the Dragon's cartridge connector

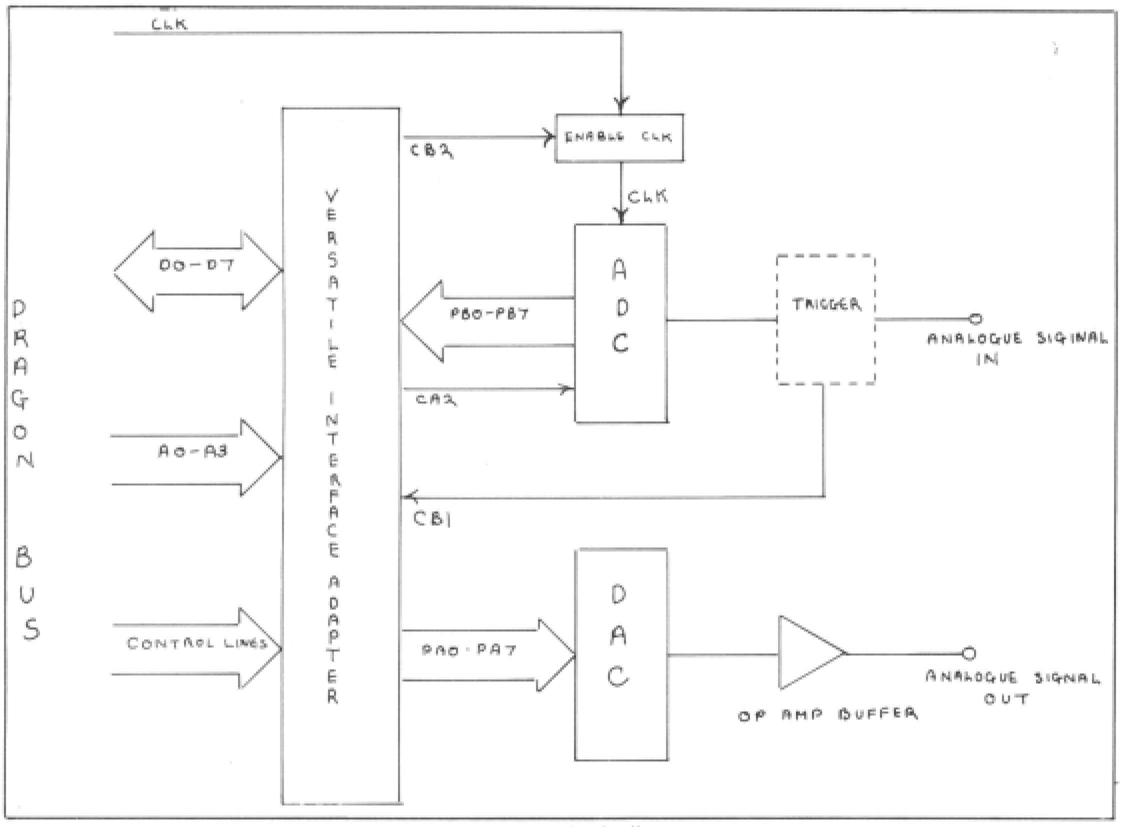


Figure 1: analogue/digital interface shown as a block diagram

APART FROM THE purely mechanical problem of getting at the cartridge connector (the address, data and control lines are accessed through this connector), interfacing the Dragon 32 is a relatively straightforward affair.

Perhaps it should be mentioned at this point that an indispensable aid to anyone wishing to interface a microcomputer is its circuit diagram. In the case of the Dragon it should be possible to obtain a copy from Dragon Data itself. The TRS-80 Color Computer *Technical Reference Manual* is also a veritable mine of information, much of which is applicable to the Dragon 32. Another source of useful information are the data sheets for the various integrated circuits used in the construction of the computer, eg the MC6809E CPU, the MC6883/74LS783 synchronous address multiplexer and the MC6847 Colour Video Display Generator.

The pinout of the Dragon's cartridge connector is shown in *Diagram 1*.

The interface itself is shown as a block diagram in *Figure 1* and in circuit form in *Figure 2*. The ZN427 ADC and the ZN426 DAC are interfaced to the Dragon 32 with the aid of a 6522 VIA (Versatile Interface Adapter). This device, as well as providing the two 8-bit parallel data ports required by the convertors, also provides the means to activate the ZN427, synchronise the clock pulses and detect any triggering signal. Because of the comparative complexity of the 6522 VIA anyone not familiar with it is recommended to obtain the relevant data sheets.

General purpose

The ADC connected to the B port of the 6522 VIA, the Ferranti ZN427, is an 8-bit successive approximation A-D connector. This is a good general purpose device, relatively cheap, easily obtained, and fairly fast. It has a 10 micro-second conversion time at a clock rate of 900 KHZ. The clock signal for the ZN427 is obtained from Pin 6 of the Dragon cartridge connector; the clock is gated to the ZN427 via a 74LS125 tri-state buffer, the purpose of which is to ensure that the incoming clock signal is synchronised to the start of conversion pulse which is obtained from the 6522 VIA's CA2 control line.

The negative voltage for the ZN427 ADC (on pin 5) may be obtained from a 7660 voltage connector IC as in the circuit illustrated; alternatively a dual voltage supply could be used. In any case it is suggested that a separate +5 Volt (VCC) supply is used for the board rather than run the risk of overloading the Dragon's PSU.

The 6522 VIA's A port feeds the DAC, a Ferranti ZN426 8-bit digital to analogue converter, the output of which is buffered by a LM358 op amp connected as a voltage follower.

For the home constructor, it is largely a matter of taste and depth of pocket how the interface is constructed. For building experimental circuits I prefer to use breadboards or plug boards. Although the intitial cost of these components is quite high, this is compensated for by the ease with which complex circuiting can be ▶

EUROSOFT

AUTO RUN

Auto Run is a utility program. It is used to add convenience and professionalism to your software.

Auto Run will help you create your title screen with the graphics editor. The graphics editor allows you to choose a background colour and border style. Using the arrow keys and several other commands you can draw pictures, block letters and also include text.

Auto Run will generate a machine language loader program to precede your program on the tape. Then, to start up your program, simply type CLOADM to load in the Auto Run loader program, which will then automatically start itself up, display your title screen, load your program and then RUN **£9.95**

TIMS

Information Management System

A user-orientated, easy to use personal database management system with these outstanding features:

- keeps files of programs, names, addresses, birthdays, recipes, class or club rosters, anything
- variable record and field lengths
- phrase substitution editor
- up to eight user-definable fields
- ML sort (up to three fields), search and delete functions
- 2 search modes — range and item
- user-definable printer format, for any printer
- up to 230 characters per record

For **£19.95** you get the database management system, our full documentation which includes a reference guide and a programmer's guide.



Inspector CLUEseau

Sherlock Holmes Agatha Cristie fans — it's finally here — a murder mystery game. Mr Goodbody has been killed in his mansion and you must solve the mystery. WHO committed the murder. WHERE did it occur and HOW was it done? Question suspects, find the secret passage, and break the code to get clues. Hi-Res graphics enhances this excellent game. The computer records the clues you obtain on a clue inventory screen and also provides suspect descriptions at the touch of a finger. A fun game that will sharpen your deductive skills. Every game is different! **£7.95**



PATTI-PAK

Two Intermission Screens (Fireworks and chase)

- extra man every 10,000 points
- 16K machine language
- invisible maze every fourth screen
- difficulty level auto-increases as you progress
- just like the arcade and more! **£7.95**

Silly Syntax



A sensational and educational version of a popular party game.

For 1 to 10 players. Load a story into the computer. The players are asked to supply a noun, verb, part of body, celebrity, etc. which the program uses to complete the story. The story, which is displayed when all words are entered, will be hilarious. For **£9.95**, you get a user guide and a tape containing the Silly Syntax game and 2 stories. You can create your own stories or order story tapes from the selection below.

Silly Syntax stories — Ten stories per tape.

SS-001 — Fairy Tales
SS-002 — Sing Along
SS-003 — X-Rated
SS-004 — Current Events
SS-006 — Adventure/Sci-Fi
SS-007 — Potpourri

Each story tape is **£7.95.**

DRAGON DISK DOCTOR

Imagine the situation that ALL disk users eventually get themselves into, namely failing to back-up the disk that just crashed. Don't panic, take two aspirins, count to 10 then call for — DISK DOCTOR.

DISK DOCTOR's features include:

- examine disk
  - text screen display of sector
  - Hi-res Ascii and hexadecimal display
  - printer Ascii and hexadecimal
- copy disk
  - copy complete disks, individual tracks and sectors
- recover file/remove protection
- list/print directory
- tape archive
- detailed tutorial and caring for disks

DISK DOCTOR is one software product that no Dragon Disk user will want to be without **£19.95**

EUROSOFT LTD

1 Punch Lane, Limeslade, Swansea
West Glamorgan, UK

Most programs available on tape or disk. (Send SAE for full list.)

● Dealer inquiries invited ●

MAKE THE MOST OF YOUR DRAGON 32



Two great books to help you tap the power of your Dragon 32.

MAKING THE MOST OF YOUR DRAGON, by Clive Gifford, leads you through programming the Dragon from first principles, and includes more than 150 complete programs, including major arcade games. **£5.95.**

DYNAMIC GAMES FOR THE DRAGON 32 — Tim Hartnell and Robert Young. Thirty magnificent games for the Dragon are lurking in this 180 page book, to make your computer behave just as you hoped it would when you bought it. Programs include MAGIC CASTLE, SNARK ATTACK, SPACE RESCUE, REVERSI/OTHELLO, CHECKERS and CUBIK'S RUBE. Just **£4.95.**

Interface Publications, Dept. DU, 9-11 Kensington High St. London W8 5NP.
I enclose £ .... Please send me the books indicated.
☐ Dynamic Games for the Dragon 32
☐ Making the most of your Dragon.
Name ....
Address ....

GRAVESEND HOME COMPUTERS

Your home computer shop for **DRAGONS,** Disc drives, printers, software, courses, books, backup and service. Mail order also available.

INSURANCE DATABASE. Customised databases for insurance agents and representatives. Basic program **£19.95.** Help with customisation to own requirements.

211 Separate Dragon Software Titles in Stock.

TEACH TYPE. A new touchtyping program from DICOsoft based on the 150 most frequently used English words. On screen prompting of finger position, speed and accuracy guide. No finger drills, type words from the start. **£6.95.**

KOKOmaths and **TEDDY.** Two programs with stimulating and colourful graphics. **TEDDY** is a counting and mental arithmetic game for youngsters. **KOKO** is a multilevel maths tester. Both programs can be amended to suit requirements. Instructions for this are included, **£6.95. TICK TOCK.** A learning the time game for under 8s. Excellent graphics, only **£5.95.**

Pay us a visit. If you need help – then just ask. Courses available in the New Year for beginners on m/c and Basic.

We are a registered Dragon service centre.

Gravesend Home Computers Ltd

39 The Terrace, Gravesend DA12 2BA

Telephone 0474-50677

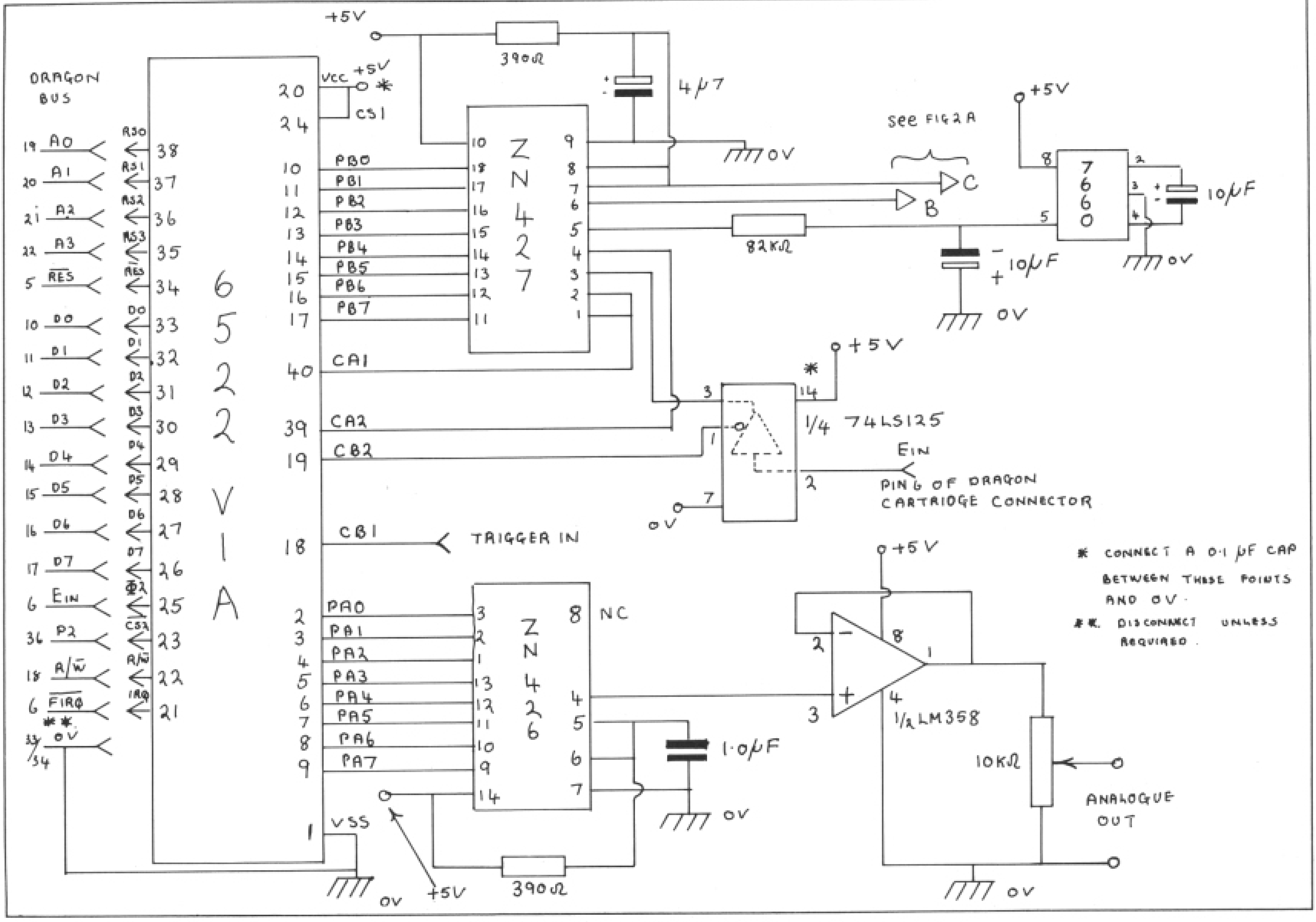


Figure 2: analogue/digital interface shown as a circuit diagram ("Trigger" is for next month's oscilloscope)

◀ assembled and altered as required.

Alternatively, the circuit can be assembled on one of the many types of Vero board available — Vero V-Q board and Vero DIP board being preferred as no track cutting is required. The DIP is very useful as it has two power rail tracks running between the IC mounting pads. Connections between the various components can be made by wire links soldered to vero pins (see *Figure 3*).

A criticism of this method is that it could give rise to problems due to stray capacitance, noise pick-up, etc, but for most applications this method has given good results. The primary requirement is the ability to produce a good soldered joint. If one of the Vero boards is used then the integrated circuits should be socketed and not soldered direct to the board.

Due precautions should be observed when handling the 6522 VIA, ZN427 and ZN426 as these may be damaged by static electricity. The circuit should also be thoroughly checked before power is applied.

Connection

For connecting interfacing circuits to the Dragon 32, I use the method shown in *Figure 4*. The extension piece was cut from double sided printed circuit board. The tracks were formed with the aid of acid resistant PCB transfers (obtainable from Maplin Electronic Supplies), sheet 10 (0 1" spaced edge connectors) and sheet 11 (straight lines 140 × 0.5m.) being required. It is extremely important that accurate register between the tracks on each side of the extension piece be maintained. Once the tracks have been etched they may be tinned, and the device given a coat of insulating varnish, with the exception of the contacting areas, of course. The interface is connected to the extension piece by means of 20 × 20 × 0 1" pitch edge connector wired to a ribbon cable.

Now on to programming and setting-up. An examination of the Dragon 32's memory map will show that the area FF00-FF5F (65280-65375) is reserved for input/output functions. Some of this space is occupied by the Dragon's two resident MC6821 peripheral interface adapters. However, any device having its chip enable or chip select line connected to P2 (Pin 36 of the Dragon Cartridge Connector) will be activated if any memory address in the area FF40-FF5F is accessed. In the circuit shown (see *Figure 2*) the 6522 VIA has its CS1 pin taken to the +5 Volt line, while its CS2 pin is taken to P2 on the Dragon Bus, the address lines A0, A1, A2, A3 are connected to the VIA's RSO, RS1, RS2, RS3 (register select) pins respectively.

In this configuration the addresses of ▶

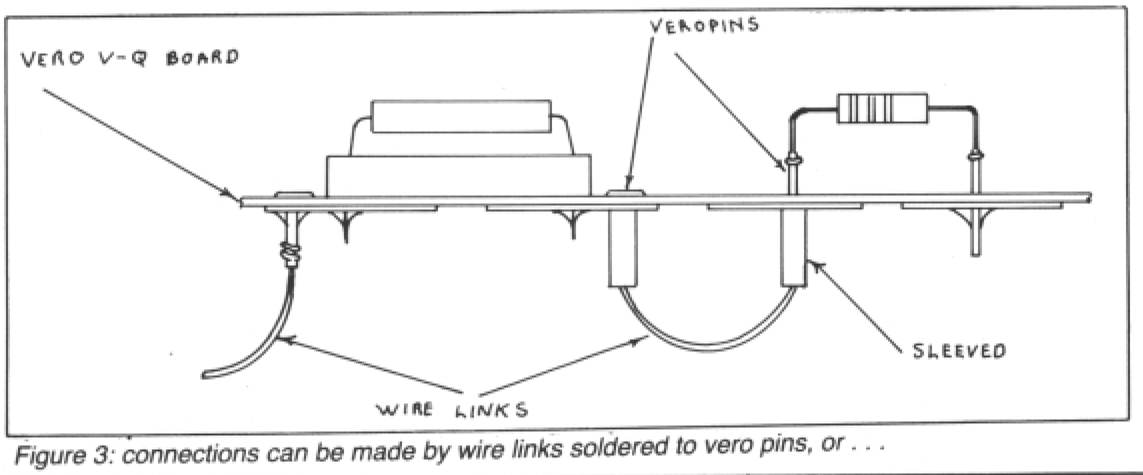
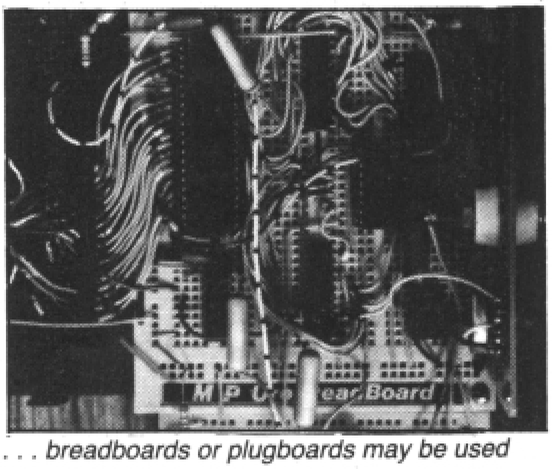


Figure 3: connections can be made by wire links soldered to vero pins, or . . .



. . . breadboards or plugboards may be used

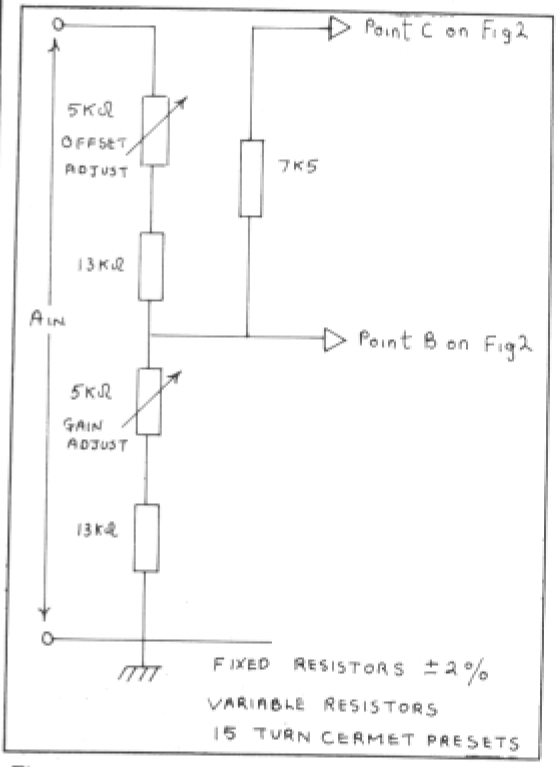


Figure 2A: bi-polar operation

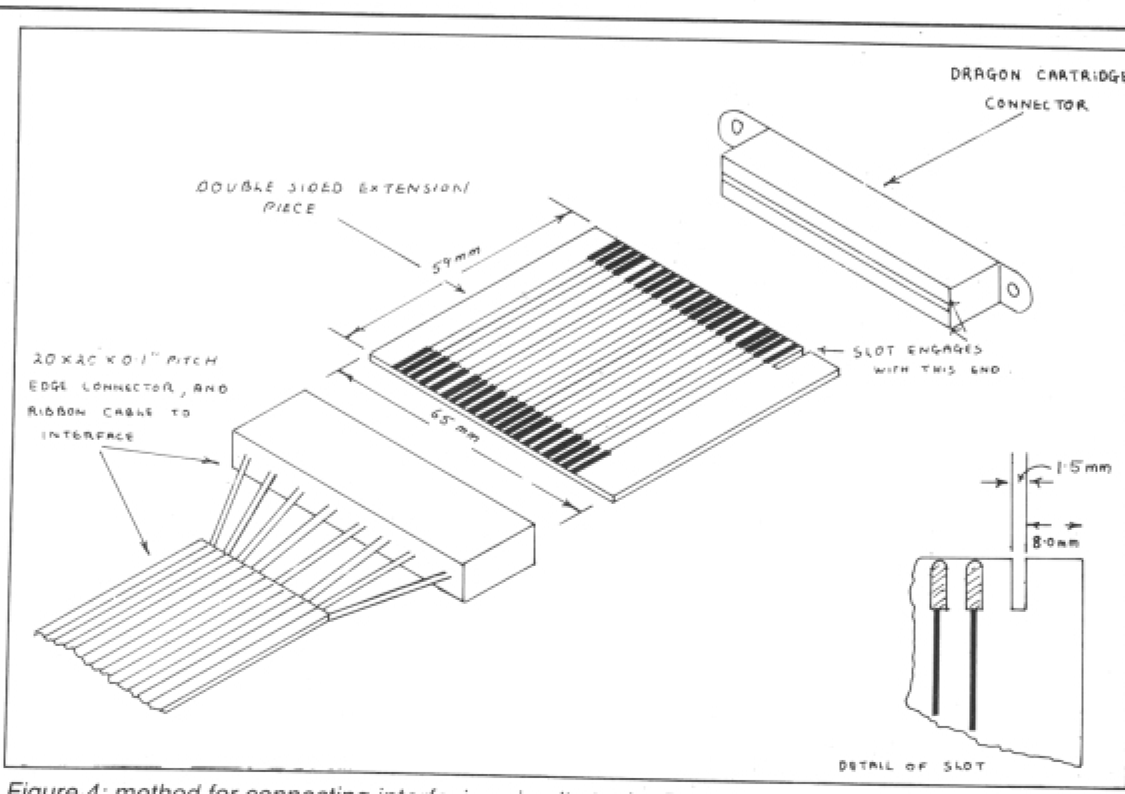


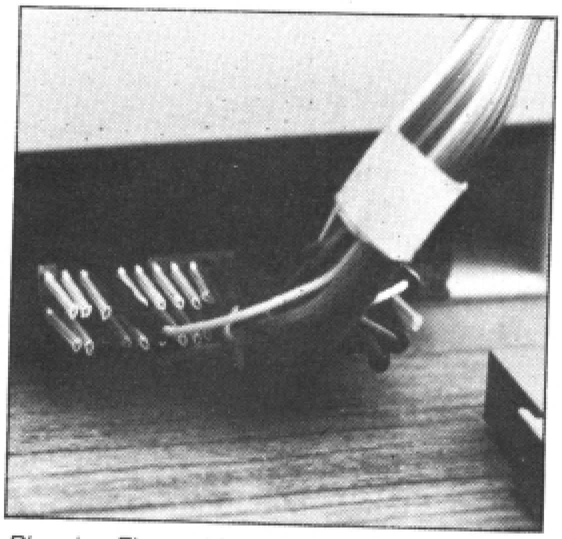
Figure 4: method for connecting interfacing circuits to the Dragon

◄ the VIA's 16 8-bit registers are as shown in *Diagram 2*. Of the 16 registers a total of six are of concern to us: 0, 1, 2, 3, 12 and 13.

First I'll look at the analogue to digital section. In order to operate this, the computer must perform the following actions:

Initialise the various registers of the 6522A Normally all registers would be initialised at the start of the program, but for the purposes of explanation each register will be dealt with as the need arises. The IRB and the ORA registers must be initialised for input and output respectively. This is accomplished by writing zero to the B Port data direction register (DDRB) at FF42, and by writing FF to the A Port data direction register (DDRA) at FF43.

Disable the clock signal to the ADC and send a start conversion pulse to it. For the ZN427 to operate correctly the clock signal must be synchronised to the start conversion pulse (see ZN427 data sheet for details). One way of doing this is to disable the clock signal while the start conversion pulse is sent to the ZN427, the clock is



Plugging Figure 4 into cartridge port

then re-enabled. Note that, except for the duration of the start conversion pulse, the WR line of the ZN427 is left high. Referring back to the circuit diagram (*Figure 2*) it will be seen that the computer clock is gated to the ZN427 via a 74LS125 tri-state buffer, whose enable line is taken to CB2 on the VIA. Also, the ZN427's Pin 4 (WR), which receives the start conversion pulse, is connected to CA2 on the VIA. The four control lines of the 65522 VIA, ie CA1, CA2, CB1 and CB2, are manipulated by writing to the peripheral control register (PCR) at address FF4C. The configuration of this register is shown in *Diagram 3*. To disable the clock CB2 is sent high by writing 111 to bits 7, 6, 5 of the PCR. This turns off the 74LS125. CA2 is put in the pulse mode by writing 101 to bits 3, 2, 1. CB1 Interrupt Control is set to active negative edge and CA1 to don't care:

POKE &HFF4, &HEB (1110 1101)

And a negative going pulse one clock cycle in length (the start conversion pulse) is sent out on CA2 by performing a write to the VIA's ORA register (see data sheet on the 6522 VIA):

POKE &HFF41,0

Re-enable clock signal The clock is re-enabled by writing 110 to bits 7, 6, 5 of the PCR, thus sending CB2 low and turning on the 74LS125. At the same time CA2 is sent high by writing 111 to bits 3, 2, 1. The CB1 Interrupt Control Line (which is connected to the trigger output) is set to respond to an active negative edge: ►

Register Number	Register Select Coding				Dragon Memory Address	Register Designation and Description
	RS3 (A3)	RS2 (A2)	RS1 (A1)	RS0 (A0)		
0	0	0	0	0	FF40	ORB/IRB Output/Input Register B
1	0	0	0	1	FF41	ORA/IRA Output/Input Register A
2	0	0	1	0	FF42	DDRB Data Direction Register B
3	0	0	1	1	FF43	DDRA Data Direction Register A
4	0	1	0	0	FF44	TIC-L T1 Low Order Latch/Counter
5	0	1	0	1	FF45	TIC-H T1 High Order Counter
6	0	1	1	0	FF46	TIL-L T1 Low Order Latch
7	0	1	1	1	FF47	TIL-H T1 High Order Latch
8	1	0	0	0	FF48	T2C-L T2 Low Order Latch/Counter
9	1	0	0	1	FF49	T2C-H T2 High Order Counter
10	1	0	1	0	FF4A	SR Shift Register
11	1	0	1	1	FF4B	ACR Auxiliary Control Register
12	1	1	0	0	FF4C	PCR Peripheral Control Register
13	1	1	0	1	FF4D	IFR Interrupt Flag Register
14	1	1	1	0	FF4E	IER Interrupt Enable Register
15	1	1	1	1	FF4F	ORA/IRA As Reg. 1 but no 'Handshake'

Diagram 2: addresses of the VIA's registers

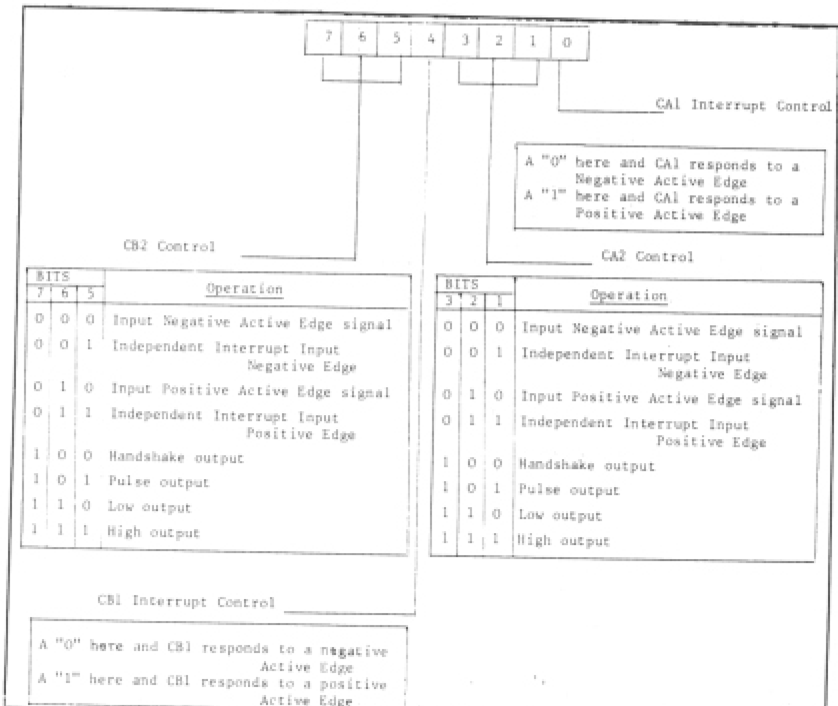


Diagram 3: peripheral control register

```
10 REM ***PROGRAM I ***
20 CLS:PRINT@6,STRING$(20,"*")
30 PRINT@38,"ADJUST BIPOLAR INPUT":PRINT@70,STRING$(20,"*")
40 PRINT@453,"PRESS BREAK TO ESCAPE"
50 POKE&HFF42,0:POKE&HFF43,&HFF
60 PRINT@99,"APPLY -4.98 VOLTS TO 'A'IN"
70 PRINT@163,"ADJUST OFFSET UNTIL READING"
80 PRINT@227,"JUST VARIES BETWEEN 0 AND 1
90 POKE&HFF4C,&HEB:POKE&HFF41,0:POKE&HFF4C,&HCF:FOR J=0 TO 10:NEXT
100 PRINT@327,"READING= ";PEEK(&HFF40)
110 TIMER=0
120 IF TIMER<50 THEN 120
130 POKE&HFF4C,&HFF
140 PRINT@335,"        "
150 A$=INKEY$:IF A$="" THEN 90 ELSE 170
160 REM***ADJUST GAIN***
170 FOR Z=1120 TO 1375:POKEZ,96:NEXT
180 PRINT@99,"APPLY +4.94 VOLTS TO 'A'IN"
190 PRINT@163,"ADJUST GAIN UNTIL READING"
200 PRINT@227,"JUST VARIES BETWEEN 254-255"
210 POKE&HFF4C,&HEB:POKE&HFF41,0:POKE&HFF4C,&HCF:FOR J=0 TO 10:NEXT
220 PRINT@327,"READING= ";PEEK(&HFF40)
230 TIMER=0
240 IF TIMER<50 THEN 240
250 POKE &HFF4C,&HFF
260 PRINT@335,"        "
270 A$=INKEY$:IF A$="" THEN 210 ELSE 290
280 REM***RE-ADJUST OFFSET***
290 FOR Z=1120 TO 1375:POKEZ,96:NEXT
300 GOTO 60
```

Program 1: for use in setting up bi-polar operation of input circuit

POKE &HFF4C, &HCF (1100 1111)

Note that the BUSY output and the RD input lines of the ZN427 are tied together, so that the tri-state outputs of the ZN427 are automatically enabled when the data is valid. These two lines are also taken to the CA1 interrupt control line, which could be used, if desired, to generate a FIRQ via the IRQ output of the 6522, when conversion is complete. However, for the purposes of this article this is not required and the IRQ output is not connected to the computer, so CA1 is set to don't care.

Process data. After a short delay (approximately 10 micro seconds!) while conversion takes place, read the B port Input Register IRB and process any data it may contain:

PEEK (&HFF40)

The next thing to look at is setting up analogue input to the ZN427. For a detailed account of connecting various ranges of analogue inputs to the ZN427 the reader is referred to the relevant data sheets. The input circuit shown in *Figure 2A* is for ±5 Volt full scale bi-polar operation. Before use this should be set up as follows:

1 Once the circuit has been thoroughly checked and all is OK, run *Program 1*.

2 Apply −4.98 Volts to A in and adjust the offset pot until the LSB output just alternates between 0 and 1 with all other bits at 0.

3 Apply +4.94 Volts to A in and adjust the gain pot until the LSB output just alternatives between 0 and 1 with all other bits at 1.

4 Repeat step 2.

Note that if the analogue signal is connected directly (via a 4 kilo-ohm resister) to pin 6 of the ZN427 then the analogue input range is between 0 volts and 2.55 volts.

Now on to digital to analogue conversion. Compared with the ZN427 ADC, interfacing the ZN426 DAC to the Dragon is quite straightforward. There are no clock signals, start conversion pulses or dual voltage supplies to worry about. The computer outputs data to the DAC through the A port of the 6522. The analogue output signal is taken from pin 4 of the ZN426 and is buffered by the LM358 voltage follower. (An alternative arrangement to the LM358 buffer I use is shown in the Ferranti data sheet on the ZN426 series.)

If the circuit is OK apply +5 volts to the converter and OP AMP. Although not shown in the circuit diagram (*Figure 2*), provision should be made to switch off the DAC section of the interface if it is not required. Set the gain control to maximum and with a Volt meter connected across the output run *Program 2*. Outputting zero should result in a reading of about 0.03 Volts (This small offset voltage can be removed by using one of the circuits illustrated in the Ferranti data sheet). Outputting 255 should result in a reading of 2.55 Volts.

Next month I'll explain how this analogue/digital interface can be used to convert the Dragon into a simple storage oscilloscope. This is where the mysterious "Trigger" in *Figure 2* comes in handy — all will be revealed. ■

```
10 REM ***PROGRAM II ***
20 CLS:PRINT@3,STRING$(26,"*")
30 PRINT@35,"CHECK OUTPUT OF ZN426 DAC."
40 PRINT@67,STRING$(26,"*")
50 POKE&HFF43,&HFF
60 POKE&HFF41,0
70 PRINT@256,"MINIMUM VALUE APPROX 0.03 VOLTS"
80 PRINT@453,"PRESS BREAK TO ESCAPE"
90 TIMER=0
100 IF TIMER<300 THEN 100
110 FOR Z=1280 TO 1311:POKEZ,96:NEXT
120 POKE&HFF41,&HFF
130 PRINT@256,"MAXIMUM VALUE APPROX 2.55 VOLTS"
140 TIMER=0
150 IF TIMER<300 THEN 150
160 FOR Z=1280 TO 1311:POKEZ,96:NEXT
170 GOTO 60
```

Program 2: to be used with a Voltmeter connected across the output

OPEN FILE FOR DRAGON USERS

Send us your Dragon programs, beginning with a general description and then explaining how the program is constructed. Take care that the listings are all bug-free, enclosing a cassette and, if possible, a printout. We pay £6 for each bug-free program published, double for the program of the month. If you have any problems with the listings, please send your queries to the appropriate author, *Dragon User*, 12-13 Little Newport Street, London WC2R 3LD.

Chess

From D Newby in Derby

THIS CHESS PROGRAM allows pawn promotion. "En passant" can be performed by moving the pawn and then placing a blank square on the pawn which should be taken.

Program notes

50-160	Draws the board.
170-220	Sets up the strings of the pieces.
230-460	Draws pieces on the board.
470-510	Stores location of each piece on the board.
550-640	Moves cursor.
650-730	Takes piece which is to be moved.
740-900	Places piece at new position.
1010-1300	Places piece at new position.
900-970	Updates location of piece in memory and checks if pawn has reached back rank.
1140-1290	Piece for promotion of pawn.

Variables

X, Y — Co-ordinates of cursor.
CX, CY References to old and new
PX, PY location of piece in memory.
C(8,8) — Variable storing piece location.
Values for above variable:

Black		White
1	Pawn	7
2	Bishop	8
3	Knight	9
4	Rook	10
5	Queen	11
6	King	12
0	empty square	

```
1 ' ****************************
2 '**                         **
3 '**      CHESS      BY      **
4 '**   DAVID NEWBY (C)1983   **
5 '**                         **
6 ' ****************************
10 DIMPC(20,20),CR(20,20)
20 CLS:PRINT@233,"<PLEASE WAIT>"
30 PMODE4,1:PCLS1
40 COLOR0,1
50 FORI=0TO176STEP22
60 LINE(I,0)-(I,176),PSET
70 LINE(0,I)-(176,I),PSET
80 NEXTI
90 J=10
100 FORI=10TO166STEP22
110 IFR=0 THENR=1:GOTO130:ELSER=0
120 PAINT(I,J)
130 NEXTI
140 J=J+22:IFJ=186 THEN170
150 IFR=1 THENR=0ELSER=1
160 GOTO100
170 RK$="U3R1U2R1U8L2U5R3D3R4U3R3D3R4U3R3D5L2D8R1D2R1D3L16"
180 KN$="E3U1E3U1H1G2L1H2U1E4BR2NR1BL2U1E2U1R6D1R3D2R1D6R1D9L17"
190 B$="E2R3U1R1U1H5U3E5F5E1H4E2F7D3G5D1R1D1R3F2L17BE2R13"
200 PN$="E2R3U1R1U1H1U1E1U1H2U1H1U1E1U1E2R1E1R2F1R1F2D1F1D1G1D1G2D1F1D1G1D1R1D1R
3F2L17"
210 Q$="U2L1U1R1U3L1U1R1U6H1U1H1U1H1R2D2R2D1F1D1F1D1F1D2R1U3E1U2E1F1D2F1D3R1U2E1
U1E1U1E1U1R2U2R2G1D1G1D1G1D6R1D1L1D3R1D1L1D2L14BU2R14BU4L14F1R1F2R1D1BR3U1R1E2R1
E1"
220 KG$="R5U1R6E1H1L6U1L5D1L6G1F1R6BU3BL4U3H1U1H1U1H1U1U1U1E2U1R3F1R1F2D1F1D2BU3
L1U2L2U2R2U2R2D2R2D2L2D2L1BD3U2E1U1E2R1E1R3D1F2D3G1D1G1D1G1D3BU3L15"
230 FORI=2TO156STEP22
240 IFPPOINT(I,152)=0 THENCOLOR1,0:ELSECOLOR0,1
250 DRAW"BM"+STR$(I)+",152"+PN$
260 PAINT(I+10,142),1,1
270 NEXTI
280 DRAW"BM2,174"+RK$
290 PAINT(12,164),1,1
300 COLOR0,1
```

Continued on page 52

```
310 DRAW"BM156,174"+RK$
320 DRAW"BM112,174"+B$
330 COLOR1,0
340 DRAW"BM46,174"+B$:PAINT(56,164),1,1
350 DRAW"BM134,174"+KN$
360 PAINT(144,164),1,1
370 PSET(140,160,0)
380 COLOR0,1:DRAW"BM24,174"+KN$
390 DRAW"BM70,174"+Q$
400 COLOR1,0:DRAW"BM96,175"+KG$:PAINT(106,160),1,1:DRAW"BR7BU5C0U2NH2NE2"
410 FORI=1TO155STEP22
420 GET(I,133)-(I+20,153),PC,G
430 PUT(I,23)-(I+20,43),PC,PRESET
440 GET(I,155)-(I+20,175),PC,G
450 PUT(I,1)-(I+20,21),PC,PRESET
460 NEXTI
470 FORI=1TO8
480 READP:C(I,1)=P:C(I,2)=1
490 FORJ=3TO6:C(I,J)=0:NEXTJ
500 READPP:C(I,7)=7:C(I,8)=PP
510 NEXTI
520 X=88:Y=88
530 GET(1,45)-(21,65),CR,G
540 SCREEN1,0
545 I$=INKEY$:IFI$="" THEN590
550 IFI$=CHR$(94) THENY=Y-22:IFY<=0 THENY=0
560 IFI$=CHR$(10) THENY=Y+22:IFY>=154 THENY=154
570 IFI$=CHR$(8) THENX=X-22:IFX<=0 THENX=0
580 IFI$=CHR$(9) THENX=X+22:IFX>=154 THENX=154
590 GET(X+1,Y+1)-(X+21,Y+21),PC,G
600 PUT(X+1,Y+1)-(X+21,Y+21),PC,PRESET:PUT(X+1,Y+1)-(X+21,Y+21),PC,PSET
610 IFPEEK(345)=223 THENGOSUB650
620 IFC(PX,PY)=1 AND PY=8 THENGOSUB1140
630 IFC(PX,PY)=7 AND PY=1 THENGOSUB1220
640 GOTO545
650 IFRT=1 THENRT=0:GOTO740
660 IFX<>0 THENCX=X/22+1ELSECX=1
670 IFY<>0 THENCY=Y/22+1ELSECY=1
680 SOUND200,1
690 GET(X+1,Y+1)-(X+21,Y+21),PC,G
700 IFPPOINT(X+1,Y+1)=1 THENPUT(X+1,Y+1)-(X+21,Y+21),CR,PSET:ELSEPUT(X+1,Y+1)-(X
+21,Y+21),CR,PRESET
710 PUT(200,0)-(220,20),PC,PSET
720 RT=1
730 RETURN
740 COLOR0,1:DRAW"BM"+STR$(X+2)+","+STR$(Y+20)
750 IFC(CX,CY)=0 THENSOUND20,1:IFPPOINT(X+1,Y+1)=1 THENPUT(X+1,Y+1)-(X+21,Y+21),
CR,PSET:GOTO910:ELSEPUT(X+1,Y+1)-(X+21,Y+21),CR,PRESET:GOTO910
760 IFPPOINT(X+1,Y+1)=0 THEN990
770 PUT(X+1,Y+1)-(X+21,Y+21),CR,PSET
780 ONC(CX,CY) GOTO790,800,810,820,830,840,850,860,870,880,890,900
790 DRAWPN$:PAINT(X+10,Y+10),0,0:GOTO910
800 DRAWB$:PAINT(X+10,Y+10),0,0:GOTO910
810 DRAWKN$:PAINT(X+10,Y+10),0,0:PSET(X+8,Y+6,1):GOTO910
820 DRAWRK$:PAINT(X+10,Y+10),0,0:GOTO910
830 DRAW"BR2"+Q$:PAINT(X+10,Y+10),0,0:PAINT(X+10,Y+19),0,0:GOTO910
840 DRAW"BR6BD1"+KG$:PAINT(X+18,Y+6),0,0:DRAW"BR7BU5C1U2NH2NE2":GOTO910
850 DRAWPN$:GOTO910
860 DRAWB$:GOTO910
870 DRAWKN$:GOTO910
880 DRAWRK$:GOTO910
890 DRAW"BR2"+Q$:GOTO910
900 DRAW"BR6BD1"+KG$
910 IFX<>0 THENPX=X/22+1ELSEPX=1
920 IFY<>0 THENPY=Y/22+1ELSEPY=1
930 IFCX=PX ANDCY=PY THEN950
940 C(PX,PY)=C(CX,CY):C(CX,CY)=0
950 PUT(200,0)-(220,20),CR,PSET
960 IFC(PX,PY)=1 AND PY=8 THENGOTO1140
970 IFC(PX,PY)=7 AND PY=1 THENGOTO1220
```

```
 980 RETURN          990 COLOR1,0          1000 PUT(X+1,Y+1)-(X+21,Y+21),CR,PRESET
1010 ONC(CX,CY) GOTO1020,1030,1040,1050,1060,1070,1080,1090,1100,1110,1120,1130
1020 DRAWPN$:GOTO910
1030 DRAWB$:GOTO910
1040 DRAWKN$:GOTO910
1050 DRAWRK$:GOTO910
1060 DRAW"BR2"+Q$:GOTO910
1070 DRAW"BR6BD1"+KG$:GOTO910
1080 DRAWPN$:PAINT(X+10,Y+10),1,1:GOTO910
1090 DRAWB$:PAINT(X+10,Y+10),1,1:GOTO910
1100 DRAWKN$:PAINT(X+10,Y+10),1,1:PSET(X+8,Y+6,0):GOTO910
1110 DRAWRK$:PAINT(X+10,Y+10),1,1:GOTO910
1120 DRAW"BR2"+Q$:PAINT(X+10,Y+10),1,1:PAINT(X+10,Y+19),1,1:GOTO910
1130 DRAW"BR6BD1"+KG$:PAINT(X+18,Y+6),1,1:DRAW"BR7BU5C0U2NH2NE2":GOTO910
1140 PE$=INKEY$
1150 IFPE$="" THEN1140
1160 CX=PX:CY=PY
1170 IFPE$="Q" THENC(CX,CY)=5:GOTO740
1180 IFPE$="R" THENC(CX,CY)=4:GOTO740
1190 IFPE$="K" THENC(CX,CY)=3:GOTO740
1200 IFPE$="B" THENC(CX,CY)=2:GOTO740
1210 GOTO1140
1220 PE$=INKEY$
1230 IFPE$="" THEN1220
1240 CX=PX:CY=PY
1250 IFPE$="Q" THENC(CX,CY)=11:GOTO740
1260 IFPE$="R" THENC(CX,CY)=10:GOTO740
1270 IFPE$="K" THENC(CX,CY)=9:GOTO740
1280 IFPE$="B" THENC(CX,CY)=8:GOTO740
1290 GOTO1220
1300 DATA4,10,3,9,2,8,5,11,6,12,2,8,3,9,4,10
```

Circles

From Edwin Aird in Newcastle-upon-Tyne

THIS COLOURFUL AND short program generates random circles of all the colours available in high-res.

```
10 PMODE 3,1:SCREEN 1,1:
   PCLS
20 X=RND (250):Y=RND
   (190):S=RND (150)
30 FOR B=1 TO S:CIRCLE
   (X,Y), B, RND (8):
   SOUNDB, 1:NEXT
40 GOTO 20
```

Drag

From Simon Rundell in Poole, Dorset

DRAG is a simulation of dragster racing in PMODE 3. Use of the joysticks is not needed (I only put them in to give you a feel of actually racing if you feel that this is not necessary simply delete lines 59 to 63).

The race is calculated on the speeds reached after every 20th of a mile. This is displayed in text and then you are returned to hi-res. There is the added danger of blowing a tyre. This kills you but you are still free to complete in the next race.

Program notes

1-8	Twiddley bits
9-19	Instructions (if required)
20-25	Sets all variables required and a few more besides
26-36	Draws the cars — GETs them into an array and draws the title (with animation too)
37-45	Draws the arena and all parts that do not need to be redrawn every time
46-87	The main part — draws everything, PUTs it all — checks if a tyre has burst or if it is time to show the speeds.
88-108	The text screen showing of all speeds, times, distances, and the current world record holder
109-114	Relays the bad news to you that one of the cars has blown up — and then carries on regardless

The program, once set, will run and run and run, till time immemorial.

```
1 REM****************************
2 REM      D  R  A  G
3 REM****************************
4 REM  BY SIMON RUNDELL
5 REM****************************
6 REM  (C) RUNDELLSOFT 1983
7 REM         4TH SEPT 1983
8 REM****************************
9 CLS:PRINT@128," WOULD YOU LIKE TO KNOW ABOUT          THE GAME (Y/N)"
10 INPUTAA$:IF AA$="Y" THEN 12
11 IF AA$="N" THEN 21
12 CLS:PRINT@0,"drag A GAME FOR THE DRAGON 32      BY SIMON RUNDELL.
```

Continued on page 56

```
(C) RUNDELLSOFT 1983"
13 PRINT@128," IN THIS,A GAME FOR TWO PLAYERS,YOU ARE THE DRIVER/PIT CREW OF  A
FUNNY CAR (A SUB-SECTION OF   DRAGSTER RACING CARS).THE OBJECTIS TO COVER THE 1/
4 OF A MILE   BEFORE YOUR OPPONENT."
14 PRINT@384," HIT [ enter ] FOR MORE":INPUTAA$
15 CLS:PRINT@0,"  THE LEFT JOYSTICK CONTROLS    THE blue CAR.THE RIGHT JOYSTICK
IS THE yellow CAR.                ON THE RESPECTIVE SIDES OF THE  SCREEN ARE DISTA
NCE INDICATORS  TO THE FLAG AND ALONG THE BOTTOMIS A GRAPHIC REPRESENTATION OF
THE SPEED"
16 PRINT"EVERY FEW SECONDS A PRINTOUT IS GIVEN OF THE SPEED,TIME  AND    REMAINI
NG DISTANCE,FROM THESE   THE LEAD IS CALCULATED.THE FIRSTONE PAST THE POST IS TH
E WINNER"
17 PRINT" HIT [enter]":INPUTAA$
18 CLS:PRINT@128,"  good luck
      YOU'LL NEED IT !!"
19 FOR NN=1TO1000:NEXTNN
20 REM*********VARIABLES********
21 DIMA(28,40):DIMB(28,40)
22 W$="GREEN":HS=180
23 S=4:S1=20:S2=130:U=30:V=100
24 N5=110:N6=65:DIST=0:KK=0
25 LA=90:LB=110:RA=156:RB=110
26 REM*************************
27 PMODE3,1:SCREEN1,0:PCLS
28 DRAW"BM128,95;S8U10R2D2U4D2L2H2G2L2U2D4U2R2D10L2U2D4U2R2F2E2R2D2U4D2L2"
29 PAINT(126,95),3,4
30 GET(110,70)-(138,110),A,G
31 PAINT(126,95),2,4:GET(110,70)-(138,110),B,G
32 FOR N6=65 TO 45 STEP-5
33 DRAW"BM30,50;C4D20R10E5U10H5NL10BM+2,+20;U20R10F5D5G5NL10F5BM+20,+0U15E5R5F5D
7NL15D8BM+20,+0U20R15D5BD5NL3NR3D10L12"
34 PUT(N5,N6)-(N5+28,N6+40),B,PSET
35 SOUND(100-N6),1
36 NEXTN6
37 REM******DRAW ARENA*********
38 PCLS
39 FOR NN=170TO 30 STEP -30
40 LINE(0,NN)-(20,NN),PSET
41 LINE(236,NN)-(256,NN),PSET
42 NEXT NN
43 DATA "C4U7R1F2E2D4G2H2L1"
44 LINE(236,20)-(256,20),PSET:LINE(0,20)-(20,20),PSET:READ A$:DRAW"BM240,20;"+A$
:DRAW"BM10,20;"+A$
45 REM*****THE FRAMEWORK*******
46 DRAW"C4"
47 DRAW"C3":LINE(0,170)-(20,170-DIST),PSET,BF:DRAW"C2":LINE(238,170)-(253,170-DI
ST),PSET,BF:DRAW"C4"
48 IF DIST=150 THEN GOSUB81
49 LINE(0,0)-(256,192),PSET,B
50 LINE(128,20)-(128,172),PSET
51 LINE(20,20)-(236,172),PSET,B
52 DRAW"BM111,1;C2D9R18U9L18C4"
53 PAINT(128,10),S,2
54 PUT(LA,LB)-(LA+28,LB+40),A,PSET
55 PUT(RA,RB)-(RA+28,RB+40),B,PSET
56 IF S=1 THEN 58
57 IF RND(10)=5 THEN SOUND1,5:S=1:TIMER=0
58 IF S=4 THEN 45
59 L1=JOYSTK(0):R1=JOYSTK(2):
60 IF L1<10 THEN LA=LA-2
61 IF L1>50 THEN LA=LA+2
62 IF R1<10 THEN RA=RA-2
63 IF R1>50 THEN RA=RA+2
64 LINE(S1,180)-(S1,192),PSET:LINE(S1-5,180)-(S1-5,192),PRESET:LINE(S1-4,180)-(S
1-4,192),PRESET:LINE(S1-3,180)-(S1-3,192),PRESET
65 LINE(S2,180)-(S2,192),PSET:LINE(S2-5,180)-(S2-5,192),PRESET:LINE(S2-4,180)-(S
```

```
2-4,192),PRESET:LINE(S2-3,180)-(S2-3,192),PRESET
66 REM*****TYRE BURST*******
67 IF RND(75) =25 THEN 68 ELSE 72
68 IF RND(2)=1 THEN PUT(LA,LB)-(LA+30,LB+30),A,PSET :D$="BLUE"ELSE PUT(RA,RB)-(R
A+30,RB+30),B,PSET:D$="YELLOW"
69 PAINT(128,5),2,2
70 FOR A=1TO1000:NEXTA
71 GOSUB109
72 REM*****MOVEMENT**********
73 S1=S1+RND(3):IF S1>125 THEN S1=125
74 S2=S2+RND(3):IF S2>235 THEN S2=235
75 LINE(25,U)-(235,U),PSET:LINE(25,U-8)-(235,U-8),PRESET
76 LINE(25,V)-(235,V),PSET:LINE(25,V-8)-(235,V-8),PRESET
77 U=U+8:IF U>175 THEN U=30:GOSUB88:DIST=DIST+30
78 V=V+8:IF V>175 THEN V=30:GOSUB88:DIST=DIST+30
79 GOTO 45
80 DATA "BM40,50;C4U10R5D5NL5F3D2BR5U10R5D5NL5D5BR5U10NR5D10R5BR5U10NR5D5NR5D5R5
BR15U10R5D10NL5BR7H5NU5F5E5NU5G5BR8U10NR5D5NR5D5R5BR5U10R5D5NL5F3D2"
81 REM*****RACE STOP********
82 PAINT(128,10),2,2
83 LINE(20,105)-(236,105),PSET
84 READB$:DRAWB$
85 IF LB<RB THEN W1=W1+1 ELSE W2=W2+1
86 FOR NN=1TO3000:NEXT
87 RESTORE:GOTO23
88 REM*****INKEY$(1)*********          *  SPEED PRINTOUT     *          **********
*************
89 S3=(S1*2)-40:S4=(S2*2)-260
90 D3=(170-DIST)*3:D4=(170-DIST)*3
91 IF SGN(S3-S4)=+1 THEN D4=D4-(S3-S4): ELSE D3=D3+(S3-S4)
92 CLS:PRINT@0,"    blue car   WINS:";W1
93 PRINT@32,"SPEED";S3;"M.P.H"
94 PRINT@64,"TIME";TIMER/128;"SEC."
95 PRINT@96,"DISTANCE";D4;"YDS LEFT"
96 IF LB<RB THEN PRINT@128," leader"
97 PRINT@256,"    yellow car  WINS:";W2
98 PRINT@288,"SPEED";S4;"M.P.H"
99 PRINT@320,"TIME";TIMER/128;"SEC."
100 PRINT@352,"DISTANCE";D3;"YDS LEFT"
101 IF S3>HS THEN HS=S3:W$="BLUE"
102 IF S4>HS THEN HS=S4:W$="YELLOW"
103 IF RB<LB THEN PRINT@384,"  leader"
104 PRINT@448,"world record:";HS;"BY ";W$
105 S5=S3-S4:IF SGN(S5)=+1 THEN LB=LB-(S5/2)
106 IF SGN(S5)=-1 THEN RB=RB+(S5/2)
107 FOR NN=1TO3000:NEXTNN
108 PMODE3,1:SCREEN1,0:RETURN
109 REM*****TYRE BURST********
110 CLS
111 PRINT@0,"A TRADGEDY HAS OCCURRED-ONE OF  THE TYRES ON THE ";D$;" CAR":PRINT"
HAS EXPLODED,SHATTERING THE CAR AND KILLING THE DRIVER INSTANTLY"
112 PRINT:PRINT"YET,IN HIS MEMORY HIS CREW WILL RACE ON.....LET US RACE AGAIN!"
113 FOR N=1TO7500:NEXT
114 RESTORE:GOTO 23
```

Weaving

From C J Evans in Swansea

WHEN THE National Eisteddfod was held in Swansea in August 1982, I wanted to produce a program to be used in the technology exhibition, and which would feature the recently launched Dragon 32. The program described here is an extensively revised version of the one which was actually used in the exhibition.

The traditional woollen mills of Wales are a great attraction to tourists, and a large number of interesting patterns can be woven on the old looms (although the designers tend to choose from a rather restricted set). In fact, a huge variety of patterns can be generated simply by assigning a value to a binary number of length 2N, so that N of the bits are equal to one, and N are equal to zero. Write this number down 2N times, to form a 2N × 2N square, and then complement all the bits in the Mth row whenever the Mth bit of the original number was a zero. Outside this basic square, repeat the pattern in both directions by reflection.

Each zero or one represents an element of the pattern, with four horizontal ▶

◀ and four vertical threads; two of each may be thought of as background colours and two as foreground or contrast. A one in the pattern represents a background element, and this is obtained by bringing the four background threads to the top and weaving them with each other, while the contrast threads are woven together separately underneath.

Conversely, a zero in the pattern means that the four contrast threads are brought to the top. The practical importance of this technique is that it allows large blocks of solid colours to be produced without the threads which are being used having to make long "jumps".

The simplest patterns use one background and one contrast colour to produce a two-colour result. More complicated patterns use a different contrast colour over the middle half of the pattern (in one direction or both). The colours are chosen from the set green, yellow, blue and red. It would be a simple matter to change the SCREEN statement, and use the colour codes 5, 6, 7 and 8 instead, but if a completely free choice from all eight colours is required, the program would have to be re-written for the low-resolution screen.

Two sizes of pattern are allowed for. Other sizes could be programmed, but it would not be so easy to fit copies of them on to the screen. The randomly-generated option occasionally produces a very striking pattern, using a set of data which one would not have expected to be any good. If you wish to make a note of the data for generating a particular pattern, the last few statements of the program enable the data to be listed. The PRINT messages have been arranged to appear on the 32-column Dragon display without any breaks in the words. This makes them look a little odd in the listing.

```
100 DIM A(16),B(103)
110 PRINT"WELSH DOUBLE TAPESTRY PATTERN   IN FOUR COLOURS.  THE
BACKGROUNDCOLOUR IS USUALLY THE SAME FOR  THE HORIZONTAL AND VER
TICAL     THREADS, BUT MAY BE DIFFERENT IF";
120 PRINT"DESIRED.  ANY OF THE CONTRAST   COLOURS MAY LIKEWISE B
E THE SAME(PUT THEM ALL EQUAL FOR A TWO-  COLOUR PATTERN).  CHOO
SE FROM   THE SET 1:GREEN, 2:YELLOW,      3:BLUE, 4:RED"
130 PRINT"THE PATTERN MAY BE MADE IN TWO  SIZES.  DO YOU WANT TH
E LARGE   ONE?"
140 INPUT T$
150 IF LEFT$(T$,1)="Y" THEN N=16 ELSE N=8
160 PRINT"DO YOU WANT A RANDOM PATTERN"
170 INPUT T$
180 IF LEFT$(T$,1)="Y" THEN 280
190 PRINT"INPUT A STRING OF";N/2;"ZEROS AND";N/2;"ONES, IN ANY O
RDER; NO SPACES ORPUNCTUATION"
200 INPUT T$
210 FOR I=1 TO N
220 A(I)=0
230 IF MID$(T$,I,1)="1"THEN A(I)=1
240 NEXT I
250 INPUT"PICK 3 COLOURS FOR VERTICAL     THREADS, BACKGROUND FI
RST       ";V1,V2,V3
260 INPUT"PICK 3 COLOURS FOR HORIZONTAL   THREADS, BACKGROUND FI
RST       ";H1,H2,H3
270 GOTO 330
280 FOR I=1TO N:A(I)=0:NEXT I
290 FOR J=1 TO N/2
300 I1=RND(N):IFA(I1)=1 THEN300
310 A(I1)=1: NEXT J
320 V1=RND(4):V2=RND(4):V3=RND(4):H1=RND(4):H2=RND(4):H3=RND(4)
330 PRINT"WHEN PATTERN IS FINISHED, CLEAR THE SCREEN BY PRESSING
 ANY KEY."
340 FOR I=1 TO 2000: NEXT I
350 PMODE 1,1:SCREEN 1,0:PCLS
360 FOR J=1 TO 8*N-1 STEP 4
370 J1=N-INT(ABS(N-J/4))
380 FOR I=1 TO 8*N-1 STEP 4
390 I1=N-INT(ABS(N-I/4))
400 IF A(I1)+A(J1)<>1 THEN V=V1:H=H1:GOTO 430
410 IF I<2*N OR I>6*N THEN V=V2   ELSE V=V3
420 IF J<2*N OR J>6*N THEN H=H2   ELSE H=H3
430 PSET(I,J,V):PSET(I+2,J,H)
440 PSET(I,J+2,H):PSET(I+2,J+2,V)
450 NEXT I: NEXT J
460 GET (0,0)-(63,63),B
470 IF N=16 THEN 510
480 PUT (0,64)-(63,127),B
490 PUT (64,0)-(127,63),B
500 PUT (64,64)-(127,127),B
510 PUT (128,0)-(191,63),B
520 PUT (0,128)-(63,191),B
530 PUT (128,128)-(191,191),B
540 GET (64,0)-(127,63),B
550 PUT (192,0)-(255,63),B
560 PUT (64,128)-(127,191),B
570 PUT (192,128)-(255,191),B
580 GET (0,64)-(63,127),B
590 PUT (128,64)-(191,127),B
600 GET (64,64)-(127,127),B
610 PUT (192,64)-(255,127),B
620 IF INKEY$=""THEN 620
630 CLS:PRINT"DO YOU WANT A LIST OF THE DATA  JUST USED?"
640 INPUT T$
650 IF LEFT$(T$,1)<>"Y" THEN 700
660 IF N=8 THEN PRINT"SMALL PATTERN"
670 IF N=16 THEN PRINT"LARGE PATTERN"
680 FOR I=1 TO N: PRINT A(I);: NEXT I: PRINT
690 PRINT V1;V2;V3,H1;H2;H3
700 PRINT"RE-RUN FOR NEW PATTERN":END
```

Store print utility

From John Tierney in Blyth

WHEN investigating the contents of store, or developing machine code programs, I often find this utility program invaluable. It displays or sends to the printer the contents of a range of bytes both in hexadecimal and character form.

In lines 70 to 120, the range and destination are set up using INPUTs. Validation is limited to whether the first number is lower than the second.

The loop controlling each line output is in lines 130 to 260. Within this FOR-..NEXT loop, there are two more FOR-..NEXT loops. The first gets the contents of eight bytes in hexadecimal (always two digits), and the second gets the contents of the same eight bytes in character form.

```
10 ' Print store in hexadecimal
20 ' and characters either to
30 ' the screen or to a printer.
40 '
50 ' c John Tierney  Nov 1983
60 '
70 CLS:PRINT@260
80 INPUT "START,END";A,B
90 IF A>B THEN 70
100 CLS:PRINT@260
```

```
110 INPUT "SCREEN(0),PRINTER(2)";C
120 IF C<>0 AND C<>2 THEN 100
130 FOR X=A TO B STEP 8
140 PRINT £-C,HEX$(X);" ";
150 FOR Y=0 TO 7
160 IF PEEK(X+Y)<16 THEN PRINT £-C,"0";
170 PRINT £-C,HEX$(PEEK(X+Y));
180 IF C=2 THEN PRINT £-C," ";
190 NEXT Y
200 PRINT £-C," ";
210 IF C=2 THEN PRINT £-C," ";
220 FOR Y= 0 TO 7
230 IF PEEK(X+Y)>122 OR PEEK(X+Y)<32 THEN PRINT£-C,".";
    ELSE PRINT£-C,CHR$(PEEK(X+Y));
240 NEXT Y
250 PRINT £-C
260 NEXT X
270 END
```

Drawing

From Kevin Murray in Edinburgh

THIS program is designed to let the user get to grips with the Dragon 32's DRAW command and to help him design and construct larger and more complex instruction combinations.

The program stores the instructions in an array so that they can be easily manipulated. If you run out of space for your instructions simply change the DIM statement in line 40 and extend the loops, etc where appropriate. The program has elementary error checking for typing mistakes but this will not deal with syntax-type errors (eg 4D instead of D4).

```
10 REM DRAW COMPOSER
20 PCLEAR4:CLEAR2500
30 CD$="BMUDLREFGHCAN1234567890+-,;"
40 DIMDS$(300)
50 CLS:INPUT"WHICH PMODE DO YOU WISH TO DRAW IN";A
60 PMODE A,1:PCLS
70 INPUT"DO YOU WISH TO WORK ON A STRING STORED ON TAPE (Y-N) ";A$
80 IF A$="N" THEN 150
90 INPUT"WHAT IS THE NAME OF THE STRING  ON TAPE (FILE NAME) ";SN$:MOTOR ON:AUDI
O ON:PRINT" POSITION THE TAPE AND PRESS    ENTER THEN PRESS PLAY ON THE    RECOR
DER":INPUTA$:AUDIO OFF
100 OPEN"I",#-1,SN$
110 FOR L=1 TO 300
120 INPUT #-1,DS$(L)
130 IF EOF(-1) THEN 140 : ELSE NEXT L
140 CLOSE #-1:DS=L:GOSUB 670
150 CLS:PRINT" THE KEYS OPERATE DRAW AS IN THEMANUAL.THE EXTRA KEYS :-[SPACE] ,[
CLEAR],[ENTER] KEYS. [SPACE]  DELETES THE LAST ENTRY;[CLEAR]  DISPLAYS THE  STRI
NG. [ENTER]   INSTRUCTS THE COMPUTER TO DRAW  THE COMMAND ENTERED"
160 FOR A=1 TO 5000:NEXT A
170 SCREEN 1,1
180 D$=INKEY$:IFD$=""THEN 180
190 IF ASC(D$)=13 THEN 240
200 IF ASC(D$)=32 THEN 270
210 IF ASC(D$)=12 THEN 290
220 IF INSTR(1,CD$,D$)=0 THEN 180
230 DS$=DS$+D$:GOTO180
240 DRAWDS$:DS=DS+1:DS$(DS)=DS$:DS$="":D$=""
250 IF DS=300 THEN 290
260 GOTO180
270 IFDS=0 THEN 180:ELSE PCLS:DS$(DS)="":DS=DS-1:GOSUB 670
280 GOTO180
290 SCREEN 0,0: CLS
300 PRINT@416,"";
310 FOR L=1 TO DS:PRINTDS$(L);" ";:NEXT L:PRINT
320 INPUT"DO YOU WISH TO CONTINUE ENTERINGDRAWING [1] OR EDIT [2] OR STOREON TAP
E [3] OR PRINT OUT [4] THESTRING OR END  THE PROGRAM [5]";SC
330 IF SC<1 OR SC>5 THEN 320
340 SC=INT(SC):ON SC GOTO 170,360,500,600,710
350 REM CAN'T GET HERE ...I HOPE !
360 PRINT"  TO EDIT THE STRING ENTER THE NUMBER OF THE COMMAND  (E.g.23  Corresp
onding to,say,D5) AND    THEN IT IS EDITED OR DELETED"
370 INPUT"WHICH COMMAND";CC
380 PRINT"IS ";DS$(CC);" THE RIGHT COMMAND (Y-N)";
390 INPUT NN$:PRINT
400 IF NN$="Y" THEN 410:ELSE 370
410 LINE INPUT"ENTER THE NEW COMMAND OR XXX TO DELETE ";ND$
420 IF ND$="XXX" THEN DS$(CC)="":ELSE DS$(CC)=ND$
430 GOSUB 670
440 GOTO 320
450 REM CAN'T GET HERE
460 REM OR HERE ......
470 REM OR HERE ......I HOPE
480 GOTO 320
490 REM SAVE DATA STRING ON TAPE
500 INPUT"WHAT IS THE NAME UNDER WHICH    THIS IS TO BE STORED";SN$
510 IF SN$="" THEN 500
520 MOTOR ON:AUDIO ON
530 INPUT"PLEASE POSITION THE TAPE AND PRESS ENTER";ZZ$:AUDIO OFF
540 PRINT"PLEASE PRESS PLAY AND RECORD ON THE TAPE RECORDER AND THEN HIT  ANY KE
Y":INPUT KE$:OPEN"O",#-1,SN$
550 FOR L=1 TO DS
560 PRINT#-1,DS$(L)
570 NEXT L
580 CLOSE #-1
590 GOTO 710
600 REM PRINT THE STRING
610 FOR L=1 TO DS
620 PRINT#-2,DS$(L);" ";
630 NEXT L
640 PRINT#-2
650 GOTO 710
660 REM DRAW THE WHOLE STRING
670 PCLS:FOR L= 1 TO DS
680 DRAW DS$(L)
690 NEXTL
700 RETURN
710 END
720 REM ****PRINTED USING THE EPSON RX-80 PRINTER IN DOUBLE STRIKE AND EMPHASIZE
D MODES****
730 REM *******************************************************
740 REM ************** WRITTEN BY K. MURRAY 15/8/83 **************
750 REM *******************************************************
```

If you've got a technical question or problem write to Brian Cadge, Dragon Answers, *Dragon User*, 12/13 Little Newport Street, London WC2R 3LD.

Dragon Answers

Interfacing recorders

I AM having difficulty in deciding what the pins are for the tape socket. I tried to wire up a 5-pin DIN plug to two 6mm diameter Jack plugs (ear and mike) with no success. Can you help?

M Clements,
Uckfield,
East Sussex

YET ANOTHER letter about the cassette interface! There have been many people asking for a recommended cassette recorder or details of the cassette socket for connecting special recorders — even reel to reel recorders in one case!

The connection is via a standard 5-pin DIN plug which is easily obtainable. The pin connections are as follows:

Pin 1 — Remote control (for motor on/off).
Pin 2 — Ground (used as one connection for both Ear and Mic).
Pin 3 — Remote control.
Pin 4 — Cassette input.
Pin 5 — Cassette output.

As both input and output use the same common ground connection, some recorders will cause feedback if both Ear and Mic leads are plugged in together. To avoid this, have only one of these plugged in at any one time.

Printing cheaply

I WOULD like to add a printer to my set-up, but I cannot afford or justify something like the Epson. The obvious choice would be an Amber but at £80 plus I still find this price rather high.

I was wondering, is it possible to run the Sinclair printer on my Dragon? I know the plugs won't fit the socket on my computer, but could you tell me, if it is possible, what adaptors or leads I will need, where can I get them and how much should I expect to pay for them?

K Nicholas,
Wantage.



A LOT of readers seem to want to run a Sinclair printer from their Dragons, the great advantage of this little printer being, of course, its cost. The disadvantage, however, is that it will not connect directly with any machine other than Sinclairs.

What is needed is a special interface which will convert the ASCII codes from the Dragon's printer output to the dot arrangements for the ZX printer. At last, a company is producing just such an interface — Printerface 2 is available for £29.95 from Microtanic Computer Systems Ltd, 16 Upland Road, London SE22 (phone 01-693 1137). This will give you a 43-character line output, as well as the usual hi-res graphics that the printer offers.

Proper timing

WHAT IS the proper way of using the Dragon's timer for achieving a delay? I have tried the following which seems to work, but is there a better way?

```
10 TIMER = 0
20 T = TIMER
30 IF T = X THEN (FUNCTION)
40 GOTO 20
```

This produces a delay of 10 to 11 seconds if X = 500.

Is there an accurate figure as to how many times the timer "pulses" per second.

D. Bateman,
Royston,
Herts.

THE MOST obvious use for the timer function is for timing, rather than for creating delays. The timer value is updated by the interrupt routine, which occurs exactly 50 times a second. Therefore, to get the time in seconds use TIMER/50. As the maximum value of Timer is 65535, this gives accurate values for up to 21 minutes before looping around to zero again, but you won't want 21-minute delay loops anyway!

A line such as that below is a good way of using the timer for a delay (S is the number of seconds which the program pauses for).

```
10 TIMER = 0
20 IF TIMER < S * 50 THEN 20
30 'rest of program
```

Lower case

I AM writing a certain application program in which it would be desirable to force lower case character input. I wondered if there was some way in which Shift 0 could be achieved from within a program without manual operation? At present I use INKEY$ and translate to lower case.

N Browne,
Portsmouth,
Hants.

AS SHIFT 0 is supposed to generate the ASCII code 18, the obvious answer would be to use ?CHR$(18) to toggle between upper and lower case. Unfortunately, this doesn't work, of course. There is a simple POKE which can be used to control alpha lock — location 329. Use the following in your program:

```
POKE 329, 255 for upper case
POKE 329 0 for lower case
```

If you poke any other number (1-254) in this address, you effectively disable lower case entry.

Scrolling sideways

I WOULD like to know if it is possible to make everything on the high resolution screen scroll sideways, and if so, how?

I have seen short machine code subroutines to do this with the text screen, and to scroll the hi-res screen up and down, but never from side to side. This would help me greatly in the programming of my Dragon.

T Harvey,
Rednal,
Birmingham.

IF YOU want to stick to Basic the only way to scroll the hi-res screen sideways is to put the whole screen in an array and use GET and PUT to move it around, for example:

```
10 PMODE 4,1:COLOR 0,1:PCLS:SCREEN 1,1
20 LINE(0,0)-(255,191),PSET,B
30 DIM A(1230):GET(0,0)-(254,191),A,G
40 PUT(1,0)-(255,191),A,PSET
50 GOTO 50
```

If you try this you will see that it does work, but is rather slow. You could speed it up by not scrolling the whole screen, or scrolling more than one pixel at a time. For most purposes this should be good enough, if, however, it is still not fast enough then you will have to dabble in machine code.

No joy on games

I HAVE a Dragon 32 and I am trying to write my own games, but I can't do so with joysticks. I know the bit about A=JOYSTK (0) or (1) or (2) or (3), but I can't figure out how to use it.

Could you please tell me how it's done?

John Corso,
London W7.

AS THE Dragon basic manual made such a mess of trying to explain the use of the JOYSTK command, it's not surprising that you are confused.

The command A=JOYSTK(n) will give A a value of between 0 and 63. n=0 for the left-right of the RIGHT joystick and n=1 for the up-down of the RIGHT joystick, similarly n=2 and 3 for the LEFT joystick. A value of 0 indicates far left or all the way up, and a value of 63 indicates far right or all the way down.

A further complication is that the values of the joystick readings are only updated when 'n' is a zero, therefore sometimes you will need to put the value of JOYSTK(0) into a dummy variable just to get the correct reading for the other values.

There is no command for reading the fire button, this is done with PEEK(65280) and is quite simple.

TROJAN

Micro Computer Software & Accessories

LIGHT PEN

DRAGON 32

Introducing a **NEW** program for the Trojan Light Pen. This new **ADVANCED** program includes the following graphics facilities:

- ★ Draw Box
- ★ Draw Circle
- ★ Draw Line
- ★ Draw Picture
- ★ Colour Fill Designated Areas
- ★ Save and load pictures to and from tape

All in Hi-Res Screen in any of four colours.
The following functions are for use on the Text Screen:

- ★ Data entry and processing
- ★ Menu selection and control
- ★ Games playing — the fascinating game of NIM included on cassette

This is a first-class program which gives hints and tips on how to write programs for the pen. Ideal for many educational uses.

A TOP QUALITY PEN PLUS A FIRST-CLASS PROGRAM FOR ONLY £11.50 including VAT.

THE BEST PEN PACKAGE AVAILABLE

Dealer enquiries welcome.
Send cheque or PO to:
Dept. D.U.8

Trojan Products, 166 Derlwyn Dunvant, Swansea SA2 7PF. Tel: (0792) 205491

DRAGON DATA DISK DRIVES



IN STOCK NOW

AVAILABLE FOR IMMEDIATE DESPATCH

SINGLE DRIVE £275 inc.
DUAL DRIVE £475 inc.

Complete with: DOS, Power Supply and Manual

DRAGON 64 £225 inc

DRAGON 32 PACKAGE

Computer, Compatible Cassette Recorder, Joysticks £199.50 inc

ONLY WHILE STOCKS LAST

DRAGON JOYSTICKS £15.50 inc p&p

Cheques and POs to:

JAY-DEE COMMUNICATIONS
182A WATER STREET
PORT TALBOT, S. WALES

Cardholder's Hotline
(0639) 895738 (24 hours)

AMERICAN EXPRESS

BARCLAYCARD VISA

Large selection of software available
CREDIT DETAILS ON REQUEST

SUNSHINE

The best books for the Dragon 32



The Working Dragon 32

A library of practical sub-routines and programs. ISBN: 0 946408 01 7

- *"There clearly is a need for books of this kind which provide more than just games" — Practical Computing, Sept 1983.*
- *It's a good one" — Personal Computer News, May 20 1983*



The Dragon Trainer

Written as a combined manual and beginners course on the power of Dragon Basic. It is aimed at the beginner and assumes no previous experience of computing. ISBN: 0 946408 09 2



Dragon 32 Games Master

Learn how to write your own top level games. ISBN: 0 94068 03 3

- *"If you can't write a half way decent game after this then it will be down to your own lack of imagination. I would recommend the Brains book as the best of this selection." Which Micro — Sept 83*



Advanced Sounds & Graphics for the Dragon Computer

All the major aspects of the sound and graphics capabilities of this machine are covered in extensive detail. ISBN: 0 946408 06 8

Plus

Sunshine Software



Cruising

Quick thinking and dexterity are required to master this high speed chase game.

Look out for the Sunshine range in W.H. Smith's, Boots, John Menzies, other leading retail chains and through our national network of bookshops and specialist stores.

Dealer enquiries:
01-734 3454.

Please send me
☐ The Working Dragon 32 at £5.95 each
☐ The Dragon Trainer at £5.95 each
☐ Dragon 32 Games Master at £5.95 each
☐ Advanced Sound & Graphics at £5.95 each
☐ Cruising at £6.95 each

I enclose cheque/postal order for £__________ made payable to: Sunshine Books: 12/13 Little Newport St., London WC2R 3LD

Name ______________________

Address ______________________

____

______________ Signature ______________

We can normally deliver in 4/5 days.

"ADVENTURES INTO IMAGINATION"

AVAILABLE FROM W.H. SMITH, JOHN MENZIES, BOOTS
AND ALL LEADING COMPUTER STORES



A spine chilling adventure ... enter via the dungeons ... navigate your way through 500 3-D rooms ... survive the swooping vampire bats ... reach the terrifying top ... confront and kill Count Kreepie, ridding the world of this Transylvanian Terror.

Can you survive the Top of the Tower? **£6.50.**

RICHARD SHEPHERD SOFTWARE

NEW DEALERS WELCOME — RING JOHN SALTHOUSE ON 06286 63531

Classified

DUST COVERS

★ DRAGON 32 & 64 ★

made-to-measure in quality natural vinyl with cut-outs for terminals and leads. Just send **£2.95** (no stamp required) to:

ALLEN ENTERPRISES
Freepost, Luton LU2 8BR

DRAWING PROGRAM designed by artist, black and white, requires joytick. Drives RX80. Copy of instructions £1.10, cassette £9.50. K. Garwell, 5 College Road, Alsager, Stoke-on-Trent ST7 2SS.

SOFTWARE STOCK CLEARANCE, almost half-price. Tapes from leading software houses. Tel: Chesterfield 2028 for information.

DRAGON PROGRAMS. All listings in this magazine supplied on quality cassette. Individually checked for correct loading and fully documented. Send cheque/PO for only £4.95 to: PLS Software Services, 13 Snows Hill, Chedgrave, Norfolk NR14 6HY.

FREE SOFTWARE EXCHANGE. Send any Dragon cassette with instructions and 75p p&p in exchange for a similar cassette. Over 50 programs in stock (Dragaon Data, Microdeal, Salamander, etc.). SAE for list and details: T. Mohsan, 24 Irving Place, Blackburn BB2 6LR.

COPYCODE will copy machine programs for instructions. Send 70p and SAE to Copycode, 117 Doncaster Lane, Woodlands, Near Doncaster, Yorkshire.

COMPUTER PROGRAMS copied (cassette based only). Printed inlay cards supplied if required. Minimum order 10 cassettes. Price from 44p. Tel: MG Copies (MIG Recordings), Burntwood 75375 (24-hour answering service).

G4BMK RADIO SOFTWARE
DRAGON 32/64
TAPE CARTRIDGE AND DISK
RTTY TRANSCEIVE. Any Baud rate, type ahead etc. Needs no TU for reception.
MORSE TRANSCEIVE. Decodes and sends morse to 150 wpm.
MORSE TUTOR. Any speed, any spacing. Pure tones.
AMTOR SYSTEM. Available soon.
Send large SAE for full details and software/hardware price list.
GROSVENOR SOFTWARE (D)
22 Grosvenor Road, Seaford
E. Sussex BN25 2BS
(0323) 893378

Breathe new fire into your Dragon with four new games from Hierosoft
Depth Charge (plus joystick)
Space Attack (plus joystick)
City Bomber
Kamikaze Starships (plus joystick)
£5.95 each inc p&p
Special offer: Any 2 games £8
Cheques and POs payable to **Hierosoft, 11 Sharlands Close, Wickford, Essex SS11 8LU**

PRINT THAT PICTURE Dragon to Acorn/Seikosha GP100A, Tandy 100/200 fast machine code, three sizes of print, colour shading, menu driven, £5. Diaassembler for Dragon to screen or printer, full disassembly including indirection, ASCII and Hex, £5. A.S.P., 6 Monks Leys Terrace, Lincoln LN2 5JG.

EPROM PROGRAMMER for most single supply eproms from 1K to 32K bytes. Powerful fill, move, examine, change, copy, verify, program, cassette save/load functions and full documentation
£79.95 (with MACE **£89.95**)
MACE EDITOR / ASSEMBLER / MONITOR cartridge with 80-page manual for machine programming the easy and professional way
£29.95
MACHINE PROGRAMMERS PACK contains the data booklets for the Dragon's four main ICs (6809, 6821, 6847 and 6883). Over 80 pages of data plus a 6809 reference card
£3.95
DRAGBUG MONITOR AND DISASSEMBLER cassette **£12.95**
6809 COMPANION machine reference book by M. James **£2.10**
SAE brings details of over 70 items for your Dragon.
ELTEK, 29 Parkside Avenue
Littlehampton, Sussex BN17 6BG

GRAPHIPRINT high resolution print routine allows normal use of Basic, text in any colour, double height, width, etc. User manual included, £9.95. Edit-32 program, text and file editor includes replace, kill, search, copy, insert and many other features. User manual included, £9.95. SAE for games and utilities list to: Willow Software, Willow Bank, Ilston, Swansea SA2 7LD.

FOR THE DRAGON 32
SPECIAL OFFER
MOVIE PRODUCER £6.95
New Original Game
Can you survive the movie business?
With 3D packaging and free red/green glasses
CHEQUE/P.O. to: **ED 40, 249 HIGH ST, STRATFORD, LONDON E15 2LS**

CASSETTE LEAD ADAPTER to cure those I/O errors when saving on your Dragon. Only £2.45, from Compton Computers, 9 Stockhill Road, Chilcompton, Somerset.

GRAPHIC WRITER. The first Dragon 32 program which writes its own programs. Graphic Writer will write a basic program from graphic displays you create. Introductory price £4.50. Write to: BR & LS, 110 Cumberland Street, Darlington, Co. Durham.

EDUCATIONAL TRADING games. "Marketday" will entertain sister, wife, grandmother, superb value £3.50. "Lemonade", similar Spectrum Candyfloss, £2.50, £5 pair. I. Butcher, 20 Heaton Close, Carleton, Blackpool FY6 7TY.

DRAGON HI-RES TEXT WRITER. Incorporate it in your own programs, on cassette, £3.95 including p&p. C.M.C. Software, 2/18 Braithwaite Road, Sparkbrook, Birmingham B11 1LB.

Here's my classified ad.

(Please write your copy in capital letters on the lines below.)

£0.20	£0.40	£0.60	£0.80
£1.00	£1.20	£1.40	£1.60
£1.80	£2.00	£2.20	£2.40
£2.60	£2.80	£3.00	£3.20
£3.40	£3.60	£3.80	£4.00
£4.20	£4.40	Please continue on a separate sheet of paper	

I make this.................................. words, at 20p per word so I enclose ....

Name..

Address..

....

....

Telephone..

Please cut out and send this form to: Classified Department, *Dragon User*, 12-13 Little Newport Street, London WC2R 3LD

Classified

DRAGON 32. Belvue Castle Adventure, £8. Find the treasures without getting caught in the traps. But first try to find the way into the castle. Very good sound effects. Will take days or weeks to solve. If you get stuck, our customer service department will help you win. Send PO/cheques to: K. Products, 10 Piermont Road, London SE22 0LN.

DRAGON 32/64 SOFTWARE
Games Pack One: 1 Poker (features full colour Hi-res Graphics). 2 Hangman (great game for all the family features full colour Hi-res Graphics). **Both on one tape . . . Price £4.99.** Listing service price 2p per line. Minimum charge £1.00. Send cheques/PO's to: **P. Perris, 88 Poplar Crescent, West Yorkshire BD18 2HH.**

AT LAST

An attractive dust cover, custom-made to fit your Dragon. Colour co-ordinated fabric exactly matches the case colour, finished off with a red taped edge and screen printed with Dragon. Looks super

Only **£3.50** inc p/p

COTSWOLD COVERS
6 MIDDLE ROW (FIRST FLOOR)
CHIPPING NORTON, OXON
Tel: 0608 412323
(for trade discount details)

COMPUTER CORRESPONDENCE COURSES

A complete course in BASIC programming which is designed specifically for the DRAGON micro-computer. It includes word processing, file-management, stock-control, games, and educational applications. Suitable for the complete beginner, hobbyist, teacher, and business person.

Cost of Course
£89.50 (incl)
Price per Module: **£8.95** (incl)
For full details write enclosing sae to:

COMPUTER CORRESPONDENCE COURSES
(Dept. D.U.) Churchway, West Ilsley
Nr Newbury BERKS RG16 0AH
Tel: (06328) 336

"COLOUR FANTASY" (for Dragon 32)

A fantasy extravaganza of continuously changing designs and merging colours, with sound and music. A must for artists, designers, or for body and mind relaxation. A 12-part menu of intricate, hypnotic or simple designs, user controlled.
£6.50

WINCO LTD
Laynes House
526-528 Watford Way, Mill Hill
London NW7 4RS

DRAGON WRITER. Mix text and hi-res. Different sizes, colours, angles. A sub-routine to begin all your programs, £3.50. Samyh Software, 9 Earlswells Drive, Cults, Aberdeen.

JOYSTICKS DRAGON JOYSTICKS

FULLY PROPORTIONAL WITH FIRE BUTTON
Now only **£12.95 per pair incl.**
+ £1 p&p

PERITRON
21 WOODHOUSE ROAD
LONDON N12 9EN

HOUSE CONTENTS insurance program, menu driven, ideal for index linked policies. Cassette £8. DSF Software, 68 Abbotsleigh Road, South Woodham Ferrers, Essex CM3 5SS.

CENTRONICS LEADS for the Dragon, 1.5m long to fit all standard Centronics interface printers, £13.50 inclusive VAT, postage. Quick despatch. Cheques payable to: High Tech Developments, 27 Quantock Road, Weston-super-Mare, Avon BS23 4DD.

DRAGON HI-RES text writer, incorporate it in your own programs. £2.80 inc p&p. ⋆ on cassette ⋆ — CMC Software, 2/18 Braithwaite Road, Sparbrook, Birmingham B11 1LB.

HAPPY 1984 COMPUTING

Personal Finance £6.50
Superb program offers: cassette filing system, automatic monthly payments list, 'help' function and excellent documentation.

Charm 3 £7.50
Print text on to the Pmode 3. Screen-define your own characters. Includes fast machine code routine to do the printing.
Comprehensive user manual.
If you use Hi-Res this is a must!
Cassette prices include post and packing

STYLISH SYSTEMS
24 PEVERIL AVENUE
SCUNTHIRPE, SOUTH HUMBERSIDE
DN17 1BE

FREE LISTS OF Dragon software still at lowest prices ever. For incredible money saving offers send SAE today to: ACS, PO Box AT 15, Antrim, County Antrim, BT41 1LT.

CHECK YOUR POOLS EASILY. Dragon tapes for Vernons or Littlewoods plans. SAE for details, John, 45 Smiths Lane, Fakenham, Norfolk.

ADVERT INDEX

Answers to Competition Corner, *Dragon User,* 12/13 Little Newport Street, London WC2R 3LD

Adding sound and speech

A double chance to win a double prize from JCB Microsystems – if you can solve ***Gordon Lee's*** *puzzle*

A NUMBER OF readers wrote in having had difficulty with the puzzle in the September edition of Dragon User. You may recall that the question involved two mathematicians, Sam and Paul, who had been given respectively, the sum and the product obtained from a throw of three dice.

After some time Paul, who had been given the product, stated that at best he could only narrow it down to one of two possibilities. Whereupon Sam, who had the sum, declared that he had narrowed it to three possibilities, but now he knew the values of the dice thrown.

Many readers tried to solve the puzzle by working out which dice throws could result in a sum obtainable in only three ways, and then tried to find the correct throw by relating these possibilities with the information given about the products. This failed to provide a unique answer. For prizes like these do you think it's going to be that easy!

Products . . .

Here is how it's done. With three dice there are throws possible with sums ranging from three (triple one) to 18 (triple six), and products from one (triple one) to 216 (triple six). In the case of the sums it can be readily seen that, with the exception of the very lowest and highest scores, there are many different possibilities. From this it would seem that Sam, who was given the sum, was at a disadvantage. This was not so — as we shall see.

Consider first the case of the *products.* First, run through all possible throws of three dice, counting the number of times that each different product is arrived at. This can be done with a simple program using the array DIM P(216). If the throws are generated in three FOR/NEXT loops A, B, and C, then Q = A*B*C: P(Q) = P(Q) + 1 will do this. When generating the throws, if A is always taken as the largest value shown on the dice, and C the smallest, it will eliminate problems caused by duplication of throws. For instance, the throw of two, four and six in any order is only counted the once.

Now, as Paul is unable to provide an answer there must be more than one set of dice throws that can form that product. Ask the computer to print out these values:

```
FOR N = 1 TO 216: IF P(N) > 1 THEN PRINT N: NEXT N
```

From this we can see that Paul must have been given either 4, 6, 8, 12, 16, 18, 20, 24, 30, 36, 48, 60, or 72 as the product.

The next step is to modify the program to print out the dice values for these products. The results should be listed in table form.

Product	Values of the dice (sums in brackets)		
4	2,2,1 (5)	4,1,1 (6)	
6	3,2,1 (6)	6,1,1 (8)	
8	2,2,2 (6)	4,2,1 (7)	
12	3,2,2 (7)	4,3,1 (8)	6,2,1 (9)
16	4,2,2 (8)	4,4,1 (9)	
18	3,3,2 (8)	6,3,1 (10)	
20	5,2,2 (9)	4,5,1 (10)	
24	4,3,2 (9)	6,2,2 (10)	6,4,1 (11)
30	5,3,2 (10)	6,5,1 (12)	
36	4,3,3 (10)	6,3,2 (11)	6,6,1 (13)
48	4,4,3 (11)	6,4,2 (12)	
60	5,4,3 (12)	6,5,2 (13)	
72	6,4,3 (13)	6,6,2 (14)	

While all this has been happening, we assume that Sam, realising that the sum that he has been given is not much use on its own, would be preparing such a table, since he might reasonably suppose that as Paul did not immediately come up with an answer, there must be more than one possibility open to someone who only knew the product.

Sam would then compare his sum with the values on the table. Therefore, once Paul announces that he has narrowed it down to *two* possibilities, *only if Sam's sum was 11* would he be able to make the statement: "I had reduced it to three possibilities, but now I can eliminate all but the correct one." The important clue that many readers missed was that Sam's statement was not made until after Paul had made his, and it was made on the strength of Sam having prepared the table of values and not solely on the sum that he had been given. Therefore the three dice thrown were: 4,4, and 3.

I hope that's put puzzled readers' minds at rest. And in case any readers are also wondering why no winner is announced this month, the answer to that is simpler. Because of Christmas this issue of the magazine had to be put together earlier than usual, before all the competition entries had come in. So you'll have to wait for the next issue to find out who's won the Dragon 64 — and how. In the meantime here's the latest competition, which makes the most of a traditional February event.

. . . and pennies

There was an unusual game at the Youth Club Valentine's Day Dance. The prizes were "penny" chews, and the object was to win as many as possible. First of all some slips of paper were numbered consecutively from 1 to 200 and each player was invited to choose one of these slips. The players had then to line up in front of a table on which were the box of chews and a card with these instructions on it:

TAKE A PENNY CHEW.
IF THE NUMBER ON YOUR SLIP OF PAPER CONTAINS AT LEAST ONE SEVEN, THEN *ADD* 7, OTHERWISE *SUBTRACT* 13.
IF YOUR NUMBER FALLS TO LESS THAN 1 YOU MUST RETIRE FROM THE GAME.
GO TO THE BACK OF THE QUEUE AND WAIT FOR YOUR NEXT TURN.

The game continued until all the players had been eliminated. The following statements were overheard: Annabel: "I had the maximum possible number of sweets" — Bryan: "Both Jane and I had 32 sweets each" — Chrissie: "I had eight sweets less than Annabel" — Daniel: "The number of sweets I received was the same as the number on my paper at the start of the game, but with the figures reversed". However, one of the statements was incorrect. **Who made it, and what numbers did the others start the game with?**

Prizes

THIS MONTH there are two sets of prizes to be won — so we're looking for two winners. Each will receive a package of software from JCB Microsystems of Bournemouth consisting of its Sound Extension Module, Speech Synthesis Module, the arcade game Basil Goes Ballooning and Basic Enhancer, a utilities program which speeds up Basic by stripping out REM statements, etc.

Rules

TO WIN the package of software you have to send in the most elegant solution to the puzzle. You must show both the answer to the competition and how to solve it with the use of a Basic program developed on your Dragon. As a tiebreaker, complete the following sentence in 15 words or less: "I want to add speech and sound to my Dragon because . . ."

Your entry must arrive at *Dragon User* by the last working day in February 1984. The names of the winners, and the solution to the puzzle, will be published in our May issue. You may only enter the competition once. Entries will not be acknowledged and we cannot enter into correspondence on the final result.

THE DAN DIAMOND TRILOGY

My name is Diamond, Dan Diamond,

and this is my story. A story of beautiful mermaids, bored robots and dank, dark dungeons. A story that started one muggy day in New York, and like the Big Apple, it's rotten to the core.



The Dan Diamond Trilogy is three separate adventure games. Each game may be played on its own, but clues may be found in the earlier adventures which may help later on. Each game comes with a lavishly illustrated 20-page case file, and hints (both helpful and misleading) which have been hidden in the illustrations.

Part I. Franklin's Tomb, in which our hero receives a mysterious plea for help which leads him to a hidden tomb and the mystery of the stargate.

Part II. Lost in Space, in which our hero finds himself stranded on a derelict spacecraft, doomed to travel endlessly through space, or find a way out.

Part III. Fishy Business, in which our hero lands on a watery planet, discovers the source of the plea for help and saves the day.

"It must be the most consistently enjoyable adventure game I've ever played"
– Computer Choice (Program of the Month) –

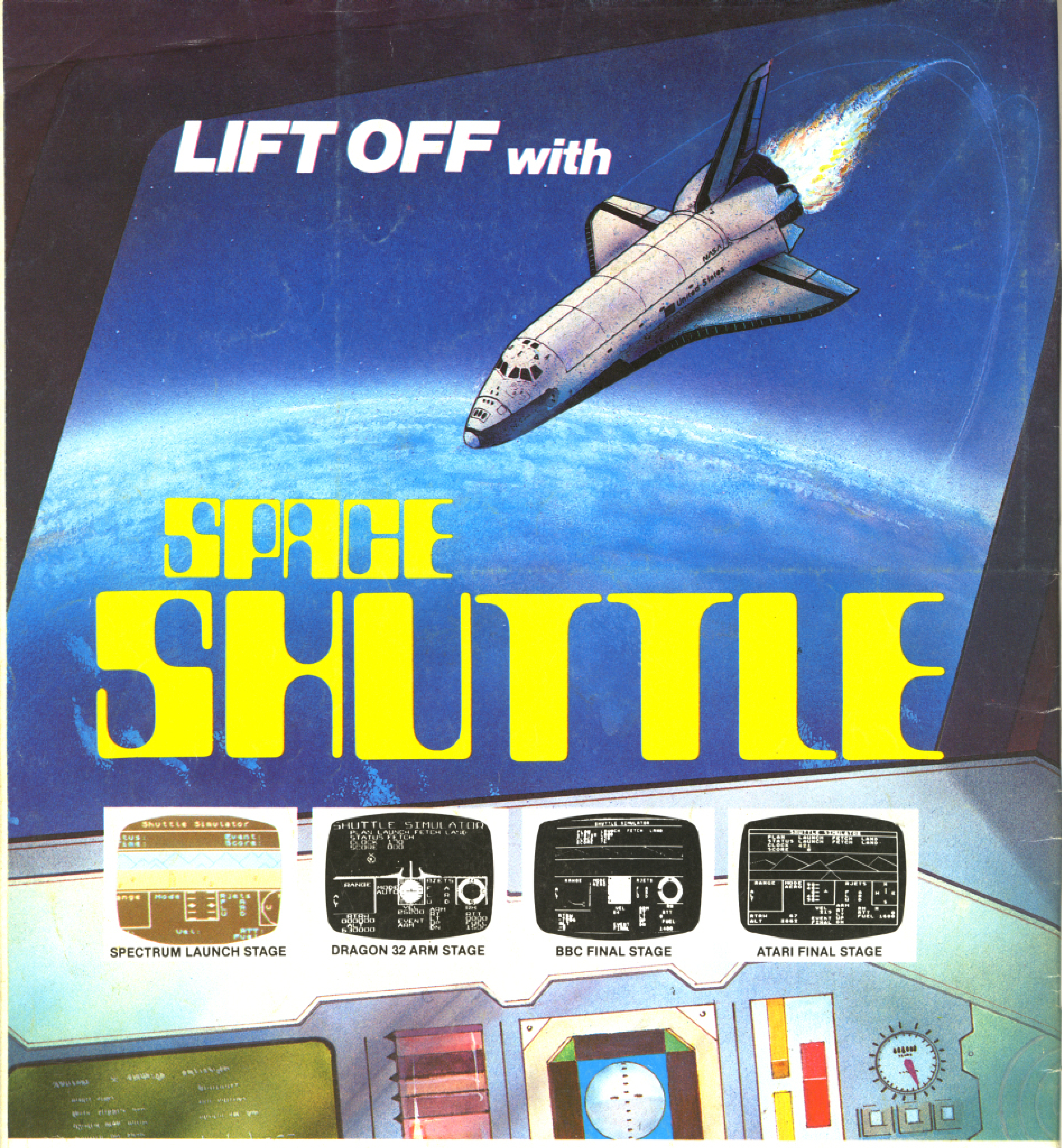
All three programs cost £9.95 each and are available for the DRAGON 32, BBC MODEL B and 48k ORIC-1 microcomputers. (note: Fishy Business for the BBC and ORIC will be available February 1984).

Cheques or postal orders payable to:

Salamander SOFTWARE

17 Norfolk Road, Brighton, East Sussex, BN1 3AA.

Look out for Dan Diamond's next Adventure Series "Franklin in Wonderland" Available Spring 1984



MISSION BRIEF

Successfully pilot the Shuttle through Launch, Fetch & Finals. Take off and position Shuttle in orbit Manoeuvre alongside malfunctioning Satellite. Retrieve - then re-enter to glide without fuel to White Sands Desert.

PREVIOUS SUCCESSFUL MISSIONS

EXCELLENT – *"Personal Computer News"*
A WELL THOUGHT OUT GAME – *"Dragon User"*
ALMOST WORTH BUYING A DRAGON FOR – *"Which Micro"*

MISSION STATUS

Available for 48K ORIC 1, ATARI 16K, DRAGON 32, BBC MODEL B, SPECTRUM 48K, COMMODORE 64, ACORN ELECTRON, TANDY COLOUR 32K

CASSETTE £8 DISK £10



Postal Orders to
41 Truro Road, St. Austell, Cornwall PL25 5JE
Telephone/Credit Card Order
Tel: 0726 3456

Available from computer dealers nationwide or from larger branches of
Boots John Menzies